

**SIMULASI SISTEM KONTROL POSISI DENGAN  
MIKROKONTROLLER AT90S8535  
SEBAGAI PENGGERAK KOLEKTOR ENERGI SURYA**

**TUGAS AKHIR**

**Sebagai Salah Satu Syarat Untuk Menyelesaikan  
Program Stratum-1 Pada Jurusan Teknik Elektro  
Fakultas Teknik Universitas Andalas**

**Oleh:**

**FAJRIL AKBAR**

**No. BP: 98 171 099**

**Pembimbing:**

**Ir. SYARKAWI SYAMSUDDIN, MSEE**

**NIP: 131 655 748**



**FAKULTAS TEKNIK  
UNIVERSITAS ANDALAS  
PADANG  
2005**

*" Dia yang menjadikan matahari bersinar dan bulan bercahaya,  
serta mengaturnya pada beberapa tempat,  
supaya kamu mengetahui bilangan tahun dan perhitungan.  
Allah tiada menciptakan demikian itu, melainkan dengan kebenaran.  
Dia terangkan beberapa tanda bagi kaum yang mau mengetahui. "*  
(Q.S. Yunus : 0 5)



*Dengan segenap rasa syukur kepada Allah SWT  
Kupersembahkan karya ini untukNya  
dan Bapakku tercinta : Dalirwar (alm),  
Mama, adikkku Ibang, Davil, Rizal, dan Rina  
Atas segala perhatian, pengorbanan dan kasih sayang yang telah diberikan selama ini.*

---

## KATA PENGANTAR

*Alhamdulillah rabbil 'alamin*, puji dan syukur penulis panjatkan kehadiran **Allah SWT** berkat limpahan rahmat dan ridho-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “**Simulasi Sistem Kontrol Posisi dengan Mikrokontroler AT90S8535 sebagai Penggerak Kolektor Energi Surya**” sebagai syarat untuk menyelesaikan program studi sarjana (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Andalas.

Selama penulisan tugas akhir ini, penulis tidak terlepas dari bantuan dan bimbingan serta pengarahan dari berbagai pihak. Untuk itu, melalui tulisan ini penulis ingin menyampaikan penghargaan dan ucapan terima kasih kepada:

1. Bapak (alm), mama dan adik-adikku atas do'a dan dukungan semangat yang tidak pernah henti-hentinya.
2. Bapak Syarkawi Syamsuddin, MSEE selaku dosen pembimbing yang telah banyak memberikan perhatian, arahan, wawasan dan nasehat yang sangat berharga kepada penulis selama penyusunan tugas akhir ini.
3. Bapak sekaligus uda Darmawan ,ST sebagai pembimbing 2 tak resmi yang telah banyak memberikan bantuan moril dan materil dalam penyelesaian tugas akhir ini.
4. Seluruh Dosen Jurusan Teknik Elektro yang telah memberikan tuntunan selama perkuliahan kepada penulis.
5. Seluruh staf adminitrasi *da Epi, da Nono, da Eka, Dodi, ni neng dan ni Pit* jurusan Teknik Elektro Fakultas Teknik Universitas Andalas.

6. *Pak Da Heru Diby Laksono, Afif, Da'i, Popo, Ocu, Rika dan Ardesriwan "pak uo"* atas bantuan dan pertolongan yang telah diberikan kepada penulis sehingga penulisan tugas akhir ini dapat diselesaikan tepat pada waktunya.
7. Seluruh sobatku di Elektro'98 tanpa terkecuali, terima kasih atas segala support, bantuan dan kebersamaan serta canda-tawa selama masa-masa kuliah
8. Bang *Jimmi "longo"*, uda *Hendriadi "gaek"* dan uda *jhen "tintin" henryco* atas kritik, saran dan diskusi-diskusi hangatnya
9. Rekan-rekan asisten LKD ; *Riki, Olin, Meri, Uun, Babe, dan Bambang*.
10. Teman-teman *Visio inTech Computindo*, terutama buat anto "*lelek*" Jr (my teacher) atas kesempatan dan pengalaman yang tak terlupakan.
11. *Maxim Integrated Products USA*, terima kasih atas bantuan komponen dan chip-nya yang sangat berguna sekali dalam penelitian ini.
12. Penulis juga mengucapkan terima kasih kepada seluruh warga HMTE lainnya serta semua pihak yang terlibat dan tidak dapat disebutkan satu persatu selama penyusunan tugas akhir ini.

Semoga Allah, SWT melimpahkan rahmat beserta karunia-Nya kepada kita semua dan semua pihak yang telah membantu penulis dalam menyelesaikan penelitian dan penyusunan tugas akhir ini. Mudah-mudahan penelitian ini bermanfaat bagi penulis dan pembaca lainnya.

Padang, Maret 2005

Penulis

## ***Abstrak***

*Penggerak kolektor energi surya adalah sebuah sistem yang akan mengarahkan kolektor surya ke arah yang memiliki intensitas cahaya matahari terbesar di berbagai tempat. Pada tugas akhir mahasiswa sebelumnya[1] telah dirancang sebuah penggerak kolektor energi surya dengan menggunakan Personal Computer (PC) sebagai pengontrol.*

*Penggunaan PC sebagai pengontrol penggerak kolektor energi surya tidak portabel karena konstruksi fisik PC yang besar. Disamping itu, banyaknya fungsi yang dijalankan sebuah PC akan memperlambat proses pengontrolan penggerak kolektor energi surya. Penggunaan mikrokontroler sebagai pengontrol dapat mengatasi semua masalah tersebut. Sebuah mikrokontroler dengan ukuran fisiknya lebih kecil dari PC dapat digunakan sebagai pengontrol energi surya tanpa banyak menjalankan fungsi-fungsi lainnya yang membuat pengontrolan akan lebih cepat dengan keakuratan yang sama dengan PC.*

*Dengan menggunakan cahaya lampu pijar sebagai sumber cahaya, maka dalam penelitian ini telah dirancang sebuah simulasi penggerak kolektor energi surya dengan menggunakan mikrokontroler AT90S8535.*

## DAFTAR ISI

<b>ABSTRAK</b> .....	i
<b>DAFTAR ISI</b> .....	ii
<b>DAFTAR GAMBAR</b> .....	iv
<b>DAFTAR TABEL</b> .....	vi

### **BAB 1. PENDAHULUAN**

Latar Belakang.....	1
Batasan Masalah.....	2
Manfaat Penelitian.....	2
Tujuan Penelitian.....	3
Sistematika Penulisan.....	3

### **BAB 2. TEORI DASAR**

2.1. Pengontrolan kumparan magnet motor DC.....	5
2.2. Spesifikasi respon transien.....	8
2.3. Faktor redaman.....	11
2.4. Konstanta penguatan motor dan konstanta waktu motor.....	17
2.5. Metode tempat kedudukan akar.....	18
2.6. Prinsip mikrokontroler.....	21
2.6.1. Unit logik aritmatik.....	22
2.6.2. Kontrol unit.....	22
2.6.3. Register–register.....	23
2.6.4. Memori.....	24
2.6.5. Input/output.....	25
2.6.6. Interupsi.....	26
2.6.7. Clock dan oscilator.....	26
2.6.8. Pengubah analog ke digital.....	26
2.6.9. Komunikasi data serial RS232.....	29
2.7. Pengubah digital ke analog.....	31

### **BAB 3 . PERANCANGAN SISTEM DAN SUBSISTEM**

3.1. Gambaran umum sistem.....	34
3.3. Perancangan modul sensor.....	37
3.2. Pemanfaatan mikrokontroler AT90S8535.....	39
3.2.1. Pemanfaatan ADC pada AT90S853.....	42
3.2.2. Pemanfaatan UART pada AT90S8535.....	45
3.4. Perancangan modul DAC dan hubungannya dengan mikrontroler....	47
3.5. Perancangan modul penguat arus penggerak motor.....	50
3.6. Perancangan modul komunikasi serial.....	52
3.7. Perancangan perangkat lunak sistem	
3.7.1. Perangkat lunak pada mikrokontroler	
3.7.1.1. Inisialisasi.....	54
3.7.1.2. Konfigurasi.....	55
3.7.1.3. Hubungan ke PC.....	56
3.7.1.4. Pengolahan data.....	57
3.7.1.5. Kirim data ke PC.....	57

### **BAB 4. PENGUJIAN SISTEM DAN SUBSISTEM**

4.1. Pengantar umum.....	59
4.2. Pengujian dan analisa modul sensor.....	59
4.3. Pengujian dan analisa karakteristik motor.....	64
4.4. Pengujian dan analisa ADC pada mikrokontroler.....	68
4.5. Pengujian dan analisa modul DAC.....	70
4.6. Pengujian dan analisa modul komunikasi serial.....	73
4.7. Pengujian sistem dan analisa.....	75

### **BAB 5. KESIMPULAN DAN SARAN**

5.1. Kesimpulan.....	82
5.2. Saran.....	82

## DAFTAR PUSTAKA

### LAMPIRAN

- Lampiran 1 ( Data pengujian sensor )
- Lampiran 2 ( Grafik pengujian km-tm motor )
- Lampiran 3 ( Data pengujian modul adc )
- Lampiran 4 ( Data pengujian modul dac )
- Lampiran 5 ( Grafik pengujian sistem )
- Lampiran 6 ( Skema rangkaian sistem )

## DAFTAR GAMBAR

Gambar 2.1. Motor DC dengan pengontrolan arus jangkar.....	5
Gambar 2.2. Spesifikasi respon transien .....	10
Gambar 2.3. Blok diagram lup tertutup sistem motor DC .....	11
Gambar 2.4. Respon redaman kritis .....	13
Gambar 2.5. Respon redaman lebih .....	14
Gambar 2.6. Letak pole pada bidang kompleks pada redaman kurang.....	14
Gambar 2.7. Respon redaman kurang .....	16
Gambar 2.8. Respon sistem dengan variasi besarnya redaman.....	17
Gambar 2.9. Sistem lup tertutup dengan penguatan K.....	20
Gambar 2.10. Penentuan tempat kedudukan akar.....	20
Gambar 2.11. Titik pisah dan titik temu.....	21
Gambar 2.12. Blok diagram umum mikrokontroler.....	21
Gambar 2.13. Blok diagram arithmetic logic unit .....	22
Gambar 2.14. Unit kontrol mikrokontroler .....	23
Gambar 2.15. Karakteristik konversi analog ke digital.....	27
Gambar 2.16. ADC tipe Successive approximation.....	28



Gambar 2.17. Sinyal transmisi rs232c.....	30
Gambar 2.18. Elemen dasar DAC .....	31
Gambar 2.19. Rangkaian konverter dengan bobot biner .....	31
Gambar 2.20. Rangkaian konverter tangga.....	32
Gambar 2.21. Karakteristik konversi digital ke analog.....	33
Gambar 3.1. Rancangan sistem.....	34
Gambar 3.2. Skema rangkaian penggerak kolektor surya.....	35
Gambar 3.3. Blok diagram sistem.....	36
Gambar 3.4. Blok diagram sistem yang disederhanakan.....	37
Gambar 3.5. Rancangan sensor.....	38
Gambar 3.6. Rangkaian pembagi tegangan pada sensor.....	38
Gambar 3.7. Blok diagram mikrokontroler AT90S8535.....	40
Gambar 3.8. Arsitektur RISC pada AT90S8535.....	40
Gambar 3.9. Peta memory AT90S8535.....	41
Gambar 3.10. Blok diagram ADC AT90S8535.....	42
Gambar 3.11. Diagram alir konversi adc dengan interupsi.....	44
Gambar 3.12. Blok diagram UART pada AT90S8535 .....	45
Gambar 3.13. Skema rangkaian modul dac .....	47
Gambar 3.14. Konfigurasi pin-pin MX7530.....	47
Gambar 3.15. Konstruksi MX7530.....	48
Gambar 3.16. Konfigurasi op-amp pada modul dac.....	48
Gambar 3.17. Rangkaian penguat penjumlah pada modul.....	49
Gambar 3.18. Rancangan penguat arus motor.....	50
Gambar 3.19. Skema rangkaian modul serial .....	51
Gambar 3.20. Hubungan serial PC dan uart AT90S8535.....	51
Gambar 3.21. Diagram alir perangkat lunak pada mikrokontroler.....	52
Gambar 4.1. Metode pengujian karakteristik sensor.....	58
Gambar 4.2. Grafik perubahan posisi sensor pada $l = 62.5$ cm.....	59
Gambar 4.3. Grafik perubahan posisi sensor pada $l = 22$ cm.....	59
Gambar 4.4. Grafik daerah kerja sistem.....	61
Gambar 4.5. Grafik persamaan masukan dari sistem.....	62
Gambar 4.6. Hubungan rangkaian pengujian respon motor.....	63

Gambar 4.7.	Diagram alir pembacaan kecepatan motor dengan PC.....	64
Gambar 4.8.	Grafik Km dan Tm motor.....	65
Gambar 4.9.	Blok diagram pengujian ADC.....	66
Gambar 4.10.	Grafik linearitas ADC kanal 0.....	67
Gambar 4.11.	Grafik linearitas ADC kanal 1.....	68
Gambar 4.12.	Skema rangkaian pengujian.....	69
Gambar 4.13.	Pengujian dengan input bit 000000001.....	70
Gambar 4.14.	Grafik linearitas DAC pada tegangan referensi 5 V.....	70
Gambar 4.15.	Diagram alir program pengujian modul serial.....	72
Gambar 4.16.	Sistem motor DC dengan konstanta K.....	75
Gambar 4.17.	Rancangan dengan konstanta K.....	75
Gambar 4.18.	Tempat kedudukan akar motor DC.....	77
Gambar 4.19.	Respon sistem saat K = 1.....	78
Gambar 4.20.	Respon sistem saat K = 2.....	78
Gambar 4.21.	Respon sistem saat K = 4.....	79
Gambar 4.22.	Karakteristik respon saat K = 2.....	79
Gambar 4.23.	Blok diagram system tanpa gangguan.....	80

### DAFTAR TABEL

Tabel 3.1.	Pemilihan kanal ADC .....	43
Tabel 3.2.	Tabel nilai setting UBRR untuk clock 8 Mhz.....	47
Tabel 4.1.	Hasil pengujian modul serial.....	74
Tabel 4.2.	Respon sistem akibat perubahan nilai K....	78

# Bab 1

## PENDAHULUAN

### 1.1. Latar belakang

Sistem kontrol posisi banyak dimanfaatkan didalam bidang industri. Salah satu aplikasi sistem kontrol posisi adalah penggerak kolektor energi surya. Penggerak kolektor energi surya adalah suatu sistem yang menggerakkan kolektor energi surya ke posisi yang memiliki intensitas cahaya matahari terbesar. Penggerak kolektor energi surya terdiri 2 buah sensor cahaya yang disusun sedemikian dengan kolektor surya dan digerakkan oleh sebuah motor.

Sistem penggerak kolektor energi surya yang telah dibuat pada tugas akhir mahasiswa sebelumnya[1] digunakan *Personal Computer* (PC) sebagai pengontrol. Penggunaan PC sebagai pengontrol menjadi kendala utama untuk aplikasi dan pemakaian penggerak kolektor surya dialam terbuka. PC dengan konstruksi fisik yang besar menyebabkan penggerak kolektor energi surya tidak portabel dan tidak praktis untuk digunakan dialam bebas. Disamping itu, banyaknya fungsi yang harus dijalankan oleh sebuah PC akan memperlambat proses pengontrolan penggerak kolektor energi surya.

Penggunaan mikrokontroler dapat mengatasi semua kendala tersebut. Mikrokontroler yang ukurannya yang kecil, lebih praktis dan portabel serta dapat menggantikan fungsi PC sebagai pengontrol. Fungsi yang dijalankan mikrokontroler sebagai pengontrol tanpa banyak menjalankan fungsi-fungsi lainnya menjadikan sistem pengontrolan lebih cepat dengan keakuratan yang sama dengan PC.

Oleh karena itulah dalam tugas akhir ini dirancang sebuah sistem penggerak kolektor energi surya dengan menggunakan mikrokontroler AT90S8535.

### 1.2. Batasan masalah

Spesifikasi penggerak yang digunakan adalah sebuah motor DC dengan pengaturan tegangan jangkar .

Rancangan sistem kontrol posisi ini, dibatasi pada kontrol posisi untuk pergerakan dua dimensi. Dengan anggapan intensitas cahaya konstan, maka pada pengujian rancangan digunakan cahaya lampu pijar sebagai sumber cahaya.

Analisa stabilitas sistem dengan metoda tempat kedudukan akar dan respon transien. Untuk mendapatkan performansi sistem yang diinginkan, maka pengaturan dilakukan dengan merubah nilai konstanta penguatan ( $K$ ).

### 1.3. Manfaat

- a. Dapat digunakan untuk menentukan arah pada peralatan yang berhubungan dengan cahaya, seperti panel pengumpul energi surya, sehingga dengan rancangan ini dapat memaksimalkan energi yang dikumpulkan.
- b. Disamping untuk mengontrol posisi peralatan yang berhubungan dengan cahaya matahari, peralatan ini dapat digunakan untuk sistem kontrol posisi lain dengan mengganti sensor cahaya.

### 1.4. Tujuan

1. Merancang dan membuat sebuah sistem penggerak kolektor energi surya dengan mikrokontroler.
2. Menganalisa pengaruh konstanta penguatan ( $K$ ) terhadap performansi sistem .  
Merekayasa sistem kontrol posisi, sehingga sistem tersebut dapat beroperasi sesuai dengan performansi yang diinginkan, yaitu respon cepat dan stabil.

### 1.5. Sistematika penulisan

#### Bab 1 Pendahuluan

Berisikan latar belakang penelitian, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan.

#### Bab 2 Teori Dasar

Membahas mengenai motor arus searah, memodelkan rancangan ke dalam model matematis, spesifikasi respon transien, faktor redaman, konstanta penguatan ( $K_m$ ), konstanta waktu ( $T_m$ ) motor, tempat kedudukan akar, prinsip mikrokontroler, pengubah analog ke digital, komunikasi data serial RS232 dan pengubah digital ke analog.

**Bab 3 Perancangan sistem beserta subsistem**

Terdiri dari penjabaran tentang sistem beserta elemen-elemen pendukungnya.

**Bab 4 Pengujian sistem beserta subsistem**

Yaitu menguji sistem dan elemen-elemen dari sistem rancangan dan membandingkan dengan literatur, sehingga dapat diketahui performansi sistem kontrol posisi yang telah dirancang.

**Bab 5 Kesimpulan dan Saran**

Merupakan kesimpulan hasil penelitian berdasarkan tujuan dan hal-hal yang diperoleh selama penelitian. Dari sini, maka akan diberikan saran-saran sehingga penelitian ini dapat lebih bermanfaat dan lebih disempurnakan di masa yang akan datang.

## Bab 2

# TEORI DASAR

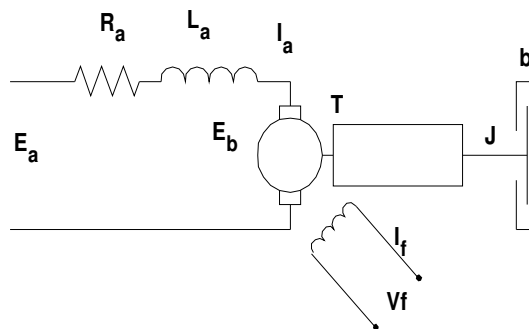
### 2.1 Pengontrolan Kumbaran Magnet Motor DC

Berdasarkan medan eksitasi motor DC terbagi atas :

1. Motor DC dengan medan eksitasi sendiri, seperti motor DC seri, motor DC shunt, motor DC kompon.
2. Motor DC dengan medan eksitasi terpisah.

Motor arus searah (DC) memiliki medan-medan yang dieksitasi secara terpisah dalam pengontrolannya dapat dibagi atas motor arus searah medan tetap dengan pengontrolan arus jangkar dan motor dc arus jangkar tetap dengan pengontrolan arus medan.

Tinjauan motor arus searah seperti terlihat pada gambar 2.1, mempunyai besaran-besaran sebagai berikut :



Gambar. 2.1 Motor DC dengan pengontrolan arus jangkar

dimana,

$R_a$  = tahanan kumbaran jangkar,  $\Omega$

$L_a$  = induksi kumbaran jangkar, *henry*

$i_a$  = arus kumbaran jangkar, *ampere*

$i_f$  = arus medan, *ampere*

$e_a$  = tegangan yang dikenakan pada kumbaran, *volt*

$e_b$  = gaya gerak listrik balik, *volt*

$\theta$  = perpindahan sudut pada poros motor, *radian*

$T$  = torsi yang diberikan oleh motor, *N-m*

$J$  = momen inersia ekivalen dari motor dan beban pada poros motor, *Kg m<sup>2</sup>*

$b$  = koefisien gesekan ekivalen dari motor dan beban pada poros motor, *N-m/rad/sec*

Torsi yang dihasilkan motor berbanding lurus dengan arus  $i_a$  dan fluks celah udara pada motor atau dapat dinyatakan dalam persamaan :

$$T = K_1 \psi i_a \quad (2-1)$$

sedangkan fluks celah udara itu sendiri adalah :

$$\psi = i_f K_f \quad (2-2)$$

dimana  $i_f$  adalah arus medan dan  $K_f$  konstanta fluks celah udara dari kedua persamaan

diatas maka didapat :  $T = i_f K_f K_1 i_a$

dan  $K = i_f K_1 K_f$  sehingga,

$$T = K I_a \quad (2-3)$$

dengan  $K_1$  sebagai konstanta dan  $K$  sebagai konstanta torsi motor

Bila kumparan medan berputar maka proses ini akan menghasilkan GGL (gaya gerak listrik) yang besarnya sebanding dengan  $K_b$  konstanta emf balik dikali dengan kecepatan sudut  $d\theta/dt$ , atau dapat dinyatakan sebagai :

$$e_b = K_b \frac{d\theta}{dt} \quad (2.4)$$

Dalam pengaturan kecepatan motor ini kita menggunakan tegangan  $e_a$  yang besarnya adalah :

$$e_a = L_a \frac{di_a}{dt} + R_a i_a + e_b \quad (2-5)$$

Arus jangkar magnet menghasilkan torsi yang bekerja terhadap inersia dan gesekan, sehingga

$$J \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = T = K i_a \quad (2-6)$$

Dengan anggapan semua persyaratan awal sama dengan nol, dan ambil persamaan laplace dari persamaan (2-4), (2-5), dan (2-6), diperoleh persamaan berikut :

$$K_b s \theta(s) = E_b(s) \quad (2-7)$$

$$(L_a s + R_a) I_a(s) + E_b(s) = E_a(s) \quad (2-8)$$

$$(J s^2 + b s) = T(s) = K I_a(s) \quad (2-9)$$

Perhatikan bahwa dengan tegangan masukan  $E_a(s)$  kita akan mendapat jumlah putaran  $\theta(s)$  dan tegangan yang terinduksi  $E_s(s)$ . Perbandingan antara tegangan jumlah putaran dengan tegangan masukan menghasilkan persamaan karakteristik sebagai berikut :

$$\frac{\theta(s)}{E_a(s)} = \frac{K}{s[L_a J s^2 + (L_a b + R_a J) s + R_a b + K K_b]} \quad (2-10)$$

Dengan anggapan nilai  $L_a \approx 0$  sehingga menghasilkan fungsi alih menjadi :

$$\frac{\theta(s)}{E_a(s)} = \frac{K_m}{s(T_m s + 1)} \quad (2-11)$$

Dimana  $T_m = (K / (R_a b + K K_b))$  disebut sebagai konstanta waktu motor

$K_m = (R_a J (R_a b + K K_b))$  konstanta penguatan motor

## 2.2 Spesifikasi Respon Transien

Karakteristik dan kinerja suatu sistem kontrol biasanya dinyatakan dalam bentuk daerah waktu. Sistem dengan komponen penyimpan energi tidak dapat merespon seketika dan menunjukkan adanya respon transien saat dikenai masukan atau gangguan.

Untuk menentukan kestabilan suatu sistem, terdapat beberapa cara yang dapat dilakukan dengan mengubah persamaan tersebut ke dalam fungsi  $s$  dengan menggunakan transformasi Laplace. Cara tersebut adalah :

- a. Menggunakan persamaan karakteristik
- b. Kriteria Routh
- c. Metode tempat kedudukan akar
- d. Kriteria Nyquist

Untuk memudahkan berbagai macam respon sistem, diasumsikan sistem mula-mula dalam keadaan diam, sehingga keluaran dan semua turunan terhadap waktu pada awal respon adalah nol.



Respon transien sistem kontrol sering menunjukkan osilasi teredam sebelum mencapai keadaan mantap (*steady state*). Dalam menentukan karakteristik transien sistem kontrol terhadap masukan tangga satuan, biasanya dicari parameter berikut [2]:

1. Lewatan maksimum (*maximum overshoot - %OS*)

Maximum *overshoot* ialah harga puncak maksimum dari kurva respon yang diukur respon yang diukur dari satu. Jika harga respon keadaan mantap tidak sama dengan satu, maka biasanya digunakan persen lewatan maksimum yang didefinisikan sebagai berikut :

$$\%OS = \frac{\text{lewatan maksimum}}{\text{nilai akhir}} \times 100\%$$

lewatan maksimum biasanya digunakan untuk menghitung kestabilan relatif dari sistem kontrol. Sistem dengan lewatan maksimum yang besar biasanya tidak diinginkan.

2. Waktu tunda (*Delay time-  $t_d$* )

Delay time dinyatakan sebagai waktu yang diperlukan oleh sebuah respon sehingga posisi respon adalah setengah dari harga akhir yang pertama kali.

3. Waktu naik (*rise time-  $t_r$* )

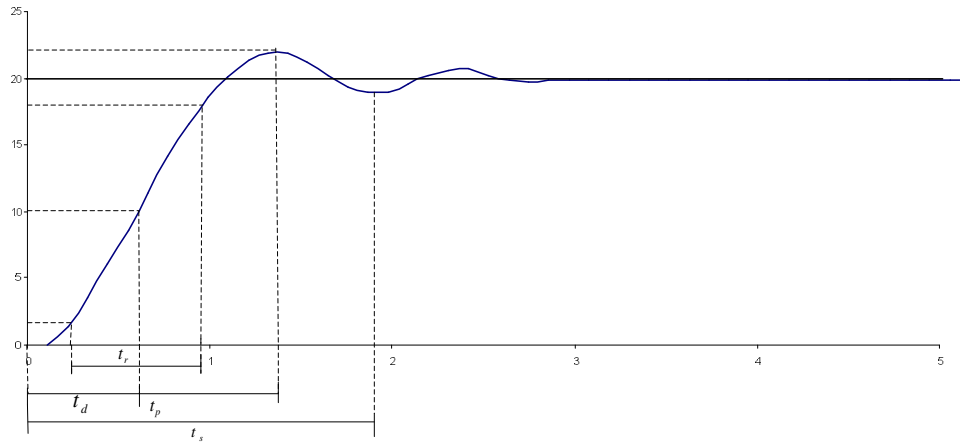
Waktu naik ialah waktu yang diperlukan respon tangga satuan untuk naik 10% sampai 90% dari harga akhirnya.

4. Waktu penetapan (*setting time-  $t_s$* )

Waktu penetapan ialah waktu yang diperlukan kurva sehingga respon tangga satuan mencapai atau menetap dalam daerah sekitar harga akhir yang ditentukan, biasanya sekitar 5%.

5. Waktu puncak (*peak time- $t_p$* )

Peak time ialah waktu yang dibutuhkan respon tangga satuan sehingga respon mencapai puncak lewatan yang pertama kalinya.



Gambar 2.2 Spesifikasi respon transien

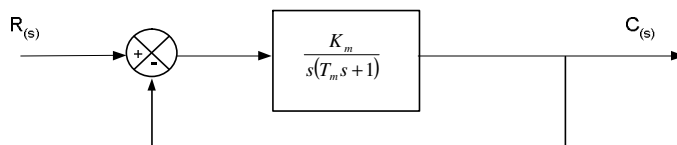
Spesifikasi daerah waktu diatas adalah cukup penting karena sebahagian besar sistem kontrol merupakan sistem daerah waktu, yang berarti sistem harus menunjukkan respon waktu yang dapat diterima atau sistem kontrol harus dimodifikasi hingga respon transiennya sesuai dengan yang diinginkan.

Dalam penerapan tertentu yang tidak mengizinkan adanya osilasi, diinginkan bahwa respon harus cukup cepat dan redaman cukup. Untuk respon transien pada orde kedua redaman  $\zeta$  harus terletak antara 0,4 dan 0,8. Harga  $\zeta < 0,4$  menghasilkan lewatan berlebih pada respon transien dan sistem dengan  $\zeta > 0,8$  akan memberikan respon lambat.

Frekwensi alami ( $\omega_n$ ) dan faktor redaman  $\zeta$  dirubah dari nol hingga mendekati satu, maka osilasi akan berkurang dan menjadi lebih teredam serta osilasi tidak akan terjadi pada saat  $\zeta$  besar dari satu.

### 2.3 Faktor Redaman

Pada dasarnya motor merupakan salah satu fungsi alih maju  $G(s)$ , sehingga dapat dinyatakan dalam blok diagram berikut:



Gambar.2.3 .Blok diagram lup tertutup sistem motor DC

Dari gambar 2.3 diatas dapat dilihat bahwa fungsi alih sistem orde dua loop tertutup adalah :

$$\frac{C(s)}{R(s)} = \frac{K_m / T_m}{s^2 + (1/T_m)s + K_m / T_m} \quad (2-30)$$

Dengan menggunakan defenisi frekwensi alami tidak teredam  $\omega_n$  dan faktor redaman  $\zeta$  , maka :

$$\omega_n^2 = \frac{K_m}{T_m} \quad (2-31)$$

dan 
$$\zeta = \frac{1}{2\omega_n T_m} \quad (2-32)$$

sehingga persamaan (2-21) dapat ditulis :

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2-33)$$

Dengan mengasumsikan keadaan awal adalah nol dan masukan  $R(s)$  berupa tangga satuan (1/s), maka persamaan diatas dapat ditulis menjadi :

$$C(s) = \frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}$$

Dengan memfaktorkan penyebutnya maka kita mendapat :

$$C(s) = \frac{\omega_n^2}{s(s + \zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1})(s + \zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1})} \quad (2-34)$$

Pada solusi sebenarnya untuk keluaran di daerah waktu tergantung pada nilai  $\zeta$  respon keluaran untuk masukan tangga satuan akan berbeda untuk setiap nilai faktor redaman, secara umum respon keluaran dapat dihitung pada faktor redaman sama dengan, lebih besar dari dan kecil dari satu.

### 1. Faktor redaman sama dengan satu ( $\zeta = 1$ )

Dengan faktor redaman sama dengan satu maka persamaan (2-16) dapat disederhanakan menjadi:

$$C(s) = \frac{\omega_n^2}{s(s + \omega_n)^2} \quad (2-35)$$

Untuk mendapatkan respon dalam waktu, maka persamaan (2-17) dengan ekspansi parsial didapatkan :

$$C(s) = \frac{K_1}{(s + \omega_n)^2} + \frac{K_2}{s + \omega_n} + \frac{K_3}{s} \quad (2-36)$$

setelah diturunkan, maka diperoleh :

$$K_1 = -\omega_n \quad K_2 = -1 \quad K_3 = 1$$

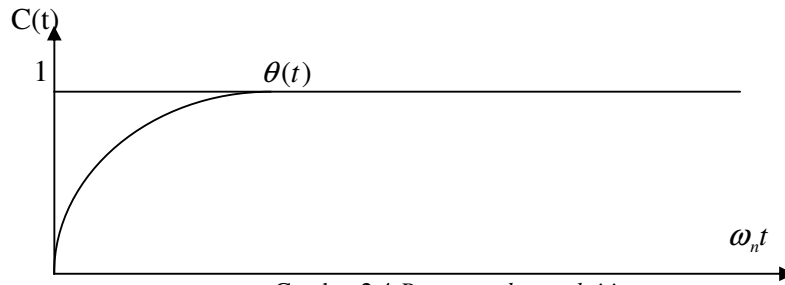
dengan mensubstitusikan K kedalam persamaan (2-18), maka dapat ditulis :

$$C(s) = \frac{\omega_n}{(s + \omega_n)^2} - \frac{1}{s + \omega_n} + \frac{1}{s} \quad (2-37)$$

dengan menggunakan tabel transformasi laplace diperoleh respon keluaran dalam daerah waktu  $C(t)$  :

$$C(t) = -\omega_n t e^{-\omega_n t} - t e^{-\omega_n t} + 1 \quad (2-38)$$

Dengan masukan tangga satuan akan membentuk respon keluaran seperti pada gambar 2.4.



Gambar 2.4 Respon redaman kritis

## 2. Faktor redaman besar dari satu ( $\zeta > 1$ )

Respon dalam daerah waktu dapat diperoleh dengan menggunakan ekspansi parsial pada persamaan (2-33), sehingga :

$$C(s) = \frac{K_1}{s} + \frac{K_2}{(s + \zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1})} + \frac{K_3}{(s + \zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1})} \quad (2-39)$$

maka  $K_1 = 1$

$$K_2 = [2(\zeta^2 - \zeta\sqrt{\zeta^2 - 1} - 1)]^{-1}$$

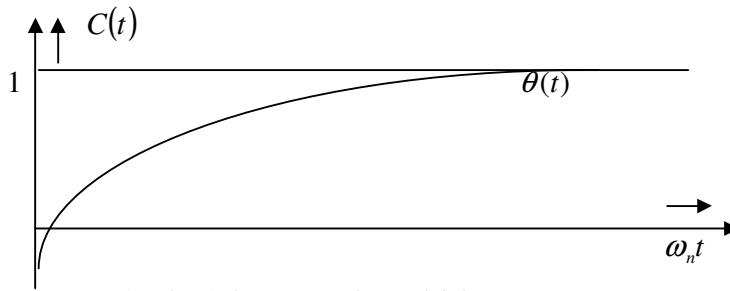
$$K_3 = [2(\zeta^2 + \zeta\sqrt{\zeta^2 - 1} - 1)]^{-1}$$

Dengan mensubstitusikan K ke persamaan (2-36), maka dapat ditulis :

$$C(s) = \frac{1}{s} + [2(\zeta^2 - \zeta\sqrt{\zeta^2 - 1} - 1)]^{-1} (s + \zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1})^{-1} \\ + [2(\zeta^2 + \zeta\sqrt{\zeta^2 - 1} - 1)]^{-1} (s + \zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1})^{-1}$$

Respon dengan keluaran pada daerah waktu diperoleh dengan menggunakan tabel transformasi laplace, sehingga dapat ditulis :

$$C(t) = 1 + \left[ 2(\zeta^2 - \zeta\sqrt{\zeta^2 - 1} - 1) \right]^{-1} e^{-(\zeta - \sqrt{\zeta^2 - 1})\omega_n t} + \left[ 2(\zeta^2 + \zeta\sqrt{\zeta^2 - 1} - 1) \right]^{-1} e^{-(\zeta + \sqrt{\zeta^2 - 1})\omega_n t} \quad (2-40)$$



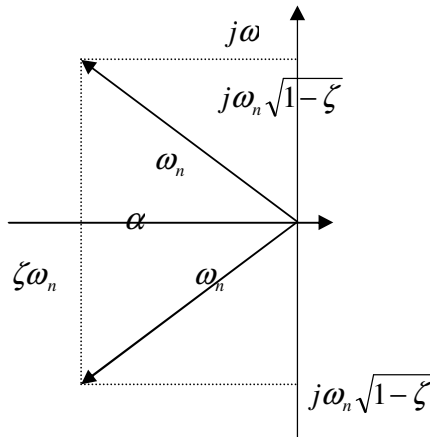
Gambar 2.5 Respon redaman lebih

### 3. Faktor redaman kurang dari 1 ( $\zeta < 1$ )

Respon dalam daerah waktu dapat diperoleh dengan melakukan ekspansi terhadap persamaan (2-33) walaupun lebih rumit karena pole pada daerah imajiner (kompleks), sehingga:

$$C(s) = \frac{K_1}{s} + \frac{K_2}{(s + \zeta\omega_n - j\omega_n\sqrt{1-\zeta^2})} + \frac{K_3}{(s + \zeta\omega_n + j\omega_n\sqrt{1-\zeta^2})} \quad (2-41)$$

untuk memudahkan menghitung  $K_1, K_2$  dan  $K_3$  maka digunakan hubungan geometris antara letak kutub imajiner pada bidang kompleks dan faktor redaman  $\zeta$  (gambar 2.6)



Gambar 2.6 Letak pole pada bidang kompleks pada redaman kurang

$$\cos \alpha = -\zeta \quad (2-42)$$

$$\sin \alpha = \sqrt{1 - \zeta^2} \quad (2-43)$$

Dengan menggunakan hubungan yang diberikan oleh persamaan (2-42) dan (2-43), maka konstanta  $K_1$ ,  $K_2$  dan  $K_3$  dapat dinyatakan dalam :

$$\begin{aligned} K_1 &= 1 \\ K_2 &= \frac{e^{-j\alpha}}{2j \sin \alpha} \\ K_3 &= -\frac{e^{-j\alpha}}{2j \sin \alpha} \end{aligned}$$

maka persamaan (2-41) dapat ditulis :

$$\begin{aligned} C(s) &= \frac{1}{s} + e^{-j\alpha} \frac{(s + \zeta\omega_n - j\omega_n\sqrt{1-\zeta^2})^{-1}}{2j \sin \alpha} \\ &\quad - e^{-j\alpha} \frac{(s + \zeta\omega_n + j\omega_n\sqrt{1-\zeta^2})^{-1}}{2j \sin \alpha} \end{aligned}$$

dengan menggunakan tabel laplace maka didapat respon keluaran dalam daerah waktu adalah :

$$C(t) = 1 + \frac{e^{-\zeta\omega_n t}}{2j \sin \alpha} e^{-(\zeta\omega_n - j\omega_n\sqrt{1-\zeta^2})t} - \frac{e^{-\zeta\omega_n t}}{2j \sin \alpha} e^{-(\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2})t}$$

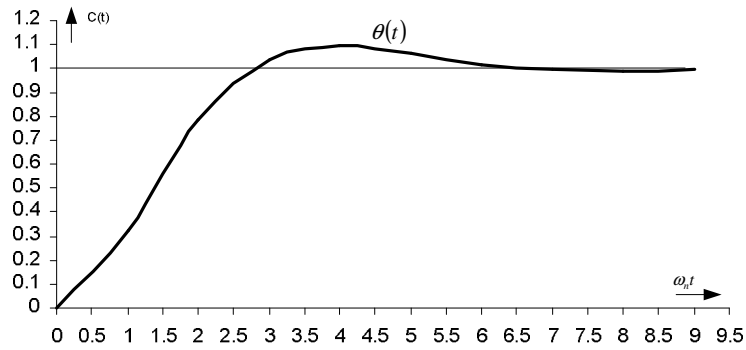
persamaan diatas dapat disederhanakan kembali menjadi :

$$C(t) = 1 + \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \frac{e^{j(\zeta\omega_n t\sqrt{1-\zeta^2}-\alpha)} - e^{-j(\zeta\omega_n t\sqrt{1-\zeta^2}-\alpha)}}{2j}$$

atau

$$C(t) = 1 + \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin(\omega_n\sqrt{1-\zeta^2}t - \alpha) \quad (2-44)$$

Respon keluaran dapat dilihat pada gambar 2.10



Gambar 2.7 Respon redaman kurang

Persamaan (2-44) menyatakan bahwa terdapat frekwensi osilasi sistem,  $\omega_m$ , yaitu:

$$\omega_m = \omega_n \sqrt{1 - \zeta^2} \quad [\text{radian}] \quad (2-45)$$

$\omega_n$  = frekwensi alami tak teredam

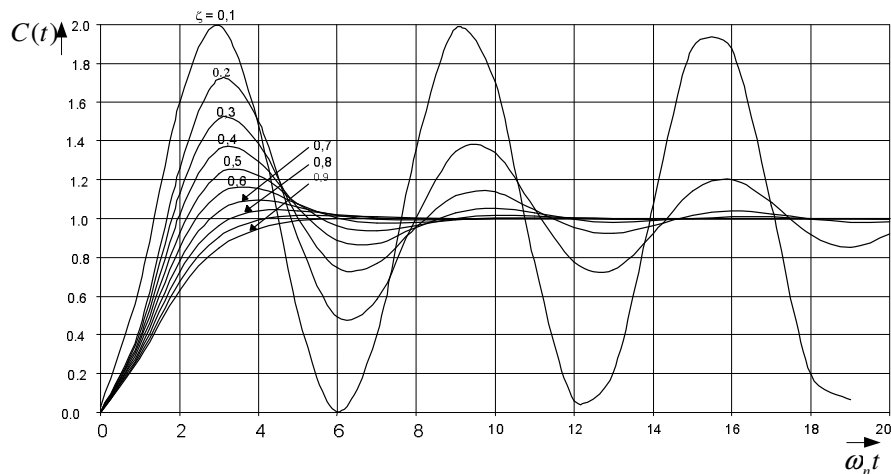
atau,

$$f_m = \frac{\omega_m}{2\pi} = \frac{\omega_n \sqrt{1 - \zeta^2}}{2\pi} \quad [\text{hertz}] \quad (2-46)$$

periode osilasi sistem,  $t_m$  adalah :

$$t_m = \frac{1}{f_m} = \frac{2\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (2-47)$$

Respon sistem kontrol orde dua dengan masukan fungsi tangga satuan dengan faktor redaman bervariasi dapat dilihat pada gambar 2.8.



Gambar.2.8 Respon sistem dengan variasi besarnya redaman

## 2.4. Konstanta Penguatan Motor ( $K_m$ ) Dan Konstanta Waktu Motor ( $T_m$ )

Untuk mendapatkan  $K_m$  dan  $T_m$  maka digunakan masukan berupa tegangan (fungsi tangga satuan) pada sistem loop terbuka, gambar 2.4

$$\theta_s = \frac{K_m T_m}{s(T_m s + 1)} \frac{E_a(s)}{s} \quad (2-48)$$

Dengan menggunakan persamaan dan tabel laplace, maka didapat fungsi waktu persamaan (2-48) adalah :

$$\theta_s = E_a (K_m t - K_m T_m + K_m T_m e^{-\frac{t}{T_m}}) \quad (2-49)$$

$$\theta^o = E_a x K_m x \left[ 1 - e^{-\frac{t}{T_m}} \right] \quad (2-50)$$

konstanta waktu adalah saat  $t = T_m$ , maka persamaan (2-41) dapat ditulis menjadi:

$$\theta^o = 0.63 x E_a x K_m \quad (2-51)$$

Dari persamaan (2-36), maka konstanta waktu adalah waktu yang dibutuhkan saat kecepatan sama dengan 0,63 dari nilai akhir.

## 2.5. Metoda tempat kedudukan akar

Gagasan utama dari tempat kedudukan akar adalah harga  $s$  dengan membuat fungsi alih loop terbuka sama dengan -1 yang harus memenuhi persamaan karakteristik sistem. Dengan metode tempat kedudukan akar didapatkan kutub loop tertutup dari kutub dan zero loop terbuka dan penguatan sebagai parameternya. Metode ini akan menghilangkan kesulitan yang dihadapi dengan metode klasik dengan memberikan peragaan grafis semua kutub loop tertutup dan harga-harga penguatan loop terbuka. Karena metode memberi paparan secara grafis maka untuk mencari akar-akar persamaan karakteristik kita dituntun untuk melakukan berbagai prosedur grafis yang efektif untuk mencari akar-akar yang polinomial yang timbul dalam studi sistem fisik.

Cara menentukan tempat kedudukan akar adalah sebagai berikut[3] :

1. Menentukan syarat sudut dan syarat besar



Pada Fungsi alih pada persamaan (2-11)

Fungsi alih loop tertutup :

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

Persamaan karakteristik loop tertutup didapat dengan menyamakan penyebut ruas kanan sama dengan nol. Atau dapat dinyatakan dengan :

$$1 + G(s)H(s) = 0$$

atau

$$G(s)H(s) = -1 \quad (2-52)$$

$G(s)H(s)$  dianggap sebagai rasio polinomial dalam  $s$ . Karena  $G(s)H(s)$  adalah besaran kompleks, maka persamaan tersebut dipisahkan menjadi dua persamaan dengan menyamakan masing-masing sudut dari kedua ruas persamaan. Untuk mendapatkan

Syarat sudut :

$$\angle G(s)H(s) = \pm 180^\circ (2k + 1) \quad (k = 0, 1, 2, \dots) \quad (2-53)$$

Syarat besaran

$$|G(s)H(s)| = 1 \quad (2-54)$$

Nilai-nilai  $s$  memenuhi syarat sudut dan syarat besaran adalah akar-akar persamaan karakteristik, atau pada loop tertutup. Suatu diagram pada bidang kompleks hanya memenuhi syarat sudut. Akar-akar dari suatu persamaan karakteristik yang merupakan besaran penguatan dapat diperoleh dengan syarat besaran.

## 2. Menghitung jumlah pole dan zero

Pada fungsi alih lup tertutup sistem pada gambar 2.13 adalah :

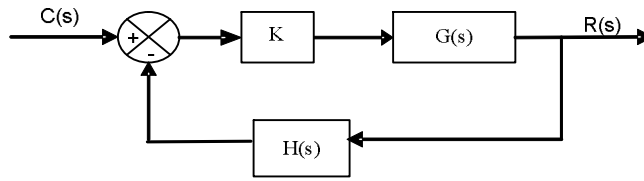
$$\frac{C(s)}{R(s)} = \frac{KG(s)}{1 + KG(s)H(s)}$$

$$\frac{C(s)}{R(s)} = \frac{KG(s)}{1 + KF(s)}$$

$$F(s) = G(s)H(s)$$

bentuk umum  $F(s)$

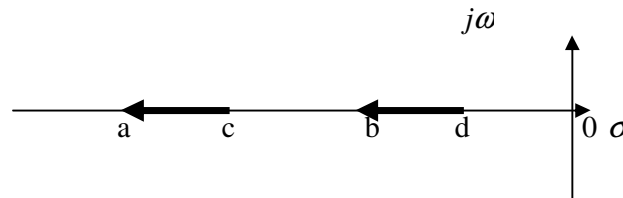
$$F(s) = \frac{(s + a)(s + b)}{(s + c)(s + d)}$$



Gambar 2.9 Sistem lup tertutup dengan penguatan K

Pole diperoleh bila F(s) sama dengan tak hingga, maka akar-akarnya adalah  $s_1 = -a$  dan  $s_2 = -b$ . Sedangkan zero diperoleh bila F(s) sama dengan nol, sehingga akar-akarnya adalah :  $s_1 = -c$  dan  $s_2 = -d$ .

Bila  $a < c < b < d$ , maka tempat kedudukan akarnya adalah :



Gambar 2.10 Penentuan tempat kedudukan akar

3. Menentukan sudut Asimtot ( $\beta$ )

$$\beta = \frac{(2k + 1)}{n - m} 180^\circ \quad (2-55)$$

$k = 0, 1, 2, \dots$

$n =$  banyak pole

$m =$  banyak zero

4. Menentukan pusat Asimtot ( $\sigma_c$ )

$$\sigma_c = \frac{\sum_{i=1}^n P_i - \sum_{i=1}^m Z_i}{n - m} \quad (2-56)$$

$\sum P_i =$  jumlah pole

$\sum Z_i =$  jumlah zero

5. Titik pisah dan titik temu

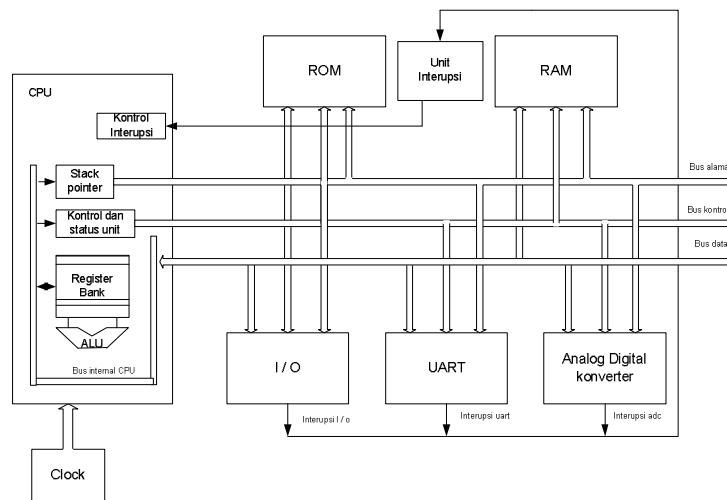


Gambar 2.11 Titik pisah dan titik temu

Metoda tempat kedudukan akar titik pisah dan titik temu dapat dilustrasikan oleh gambar (2.11) .

## 2.6. Prinsip mikrokontroler

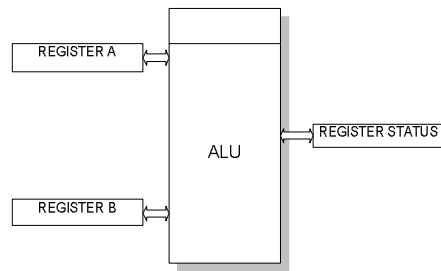
Mikrokontroler merupakan sistem yang dipaketkan dalam satu keping integrated circuit (IC). Terdapat memori, timer, counter dan I/O dalam kepingannya (*onchip*). Antarmuka (*interface*) yang menghubungkannya terintegrasi, sehingga waktu akses setiap fungsi sangat kecil dan cepat. Gambar 2.12 menunjukkan sebuah mikrokontroler dengan elemen-elemen pendukungnya.



Gambar 2.12 . Blok diagram umum mikrokontroler

### 2.6.1 Unit Logik Aritmatik (ALU)

Unit ini mendukung pelaksanaan operasi aritmatik, logika dan manipulasi bit pada mikrokontroler. Perintah-perintah yang melibatkannya antara lain penambahan, pengurangan, pergeseran register dan nilai, dan operasi penambahan logika. Arithmetic Logic Unit (ALU) tidak mengeksekusi perintah serta tidak digunakan untuk membaca dan menyimpan data atau instruksi.

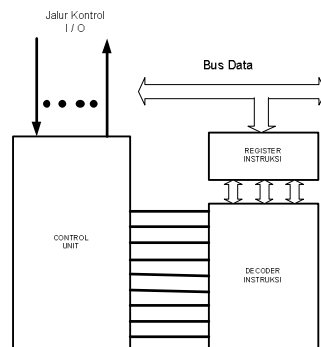


Gambar 2.13 Blok diagram arithmetic logic unit

ALU merupakan blok yang menggabungkan dua register dan melakukan set terhadap bit-bit status terkait. Gambar 2.13 memperlihatkan skema blok. Hasil dari proses ditempatkan pada register. Beberapa jenis mikrokontroler menggunakan register khusus sebagai register sumber dan register hasil. Register ini disebut akumulator.

### 2.6.2. Kontrol Unit (*Control unit*)

Unit kontrol merupakan unit yang mengatur kerja bagian-bagian lain didalam mikrokontroler, misalnya memori, I/O dan bus seperti terlihat pada gambar 2.14. Dekoder instruksi yang dimilikinya mengambil perintah dan melakukan operasi-operasi yang diwakili oleh perintah tersebut.



Gambar 2.14 Unit kontrol mikrokontroler

### 2.6.3 Register-register (register Array)

Register-register membantu mikrokontroler dalam memindahkan data dan melakukan proses proses logika. Jumlah dan lebarnya tergantung pada jenis mikrokontroler.

*Program counter* merupakan register yang memuat alamat intruksi akakn dijalankan selanjutnya. Ukurannya tergantung pada besar alamat memori yang bisa diakses oleh

mikrokontroler. Peranannya sangat besar dalam mengatur intruksi yang akan dijalankan, terutama pada rutin yang melakukan loncatan-loncatan alamat program sesuai kebutuhan aplikasi mikrokontroler.

Status register memuat bit-bit yang menyatakan keadaan mikrokontroler setiap saat, yang juga disebut sebagai *flag register*. Bit-bit ini dapat berubah sesuai dengan kerja yang dilakukan mikrokontroler. Banyak intruksi mikrokontroler yang tergantung pada register status diantaranya pencabangan dengan kondisi.

*Stack Pointer* merupakan register yang digunakan untuk menentukan alamat terakhir lokasi yang digunakan untuk penyimpanan sementara. Penyimpanan dan pengeluaran nilai pada alamatnya dilakukan diawal dan akhir sebuah blok rutin. Urutan penyimpanan dan pengambilan nilai dilakukan dengan metode *last in first out*(LIFO). Perintah yang terakhir disimpan merupakan perintah yang pertama kali dikeluarkan.

*General purpose register* merupakan register yang dapat digunakan untuk berbagai tugas yang dikehendaki. Tidak ada perintah mikrokontroler yang secara khusus menggunakan register-register ini.

#### **2.6.4. Memori**

Program dan data pada umumnya memiliki ukuran yang cukup besar dan tidak mungkin disimpan di register. Untuk itu digunakan memori sebagai penyimpanan sementara dan tetap. Memori umumnya berwujud keping IC. Terdapat dua jenis memori, yaitu *volatile memory* dan *non-volatile memory*.

1. Volatile memory

memory jenis ini kehilangan data yang disimpannya saat tidak ada catu daya, dikenal juga dengan nama memory sistem atau RAM (*Random Acces Memory*). Penyimpanan data dilakukan per *byte* dan setiap lokasi memiliki alamat spesifik. Proses menulis dan membaca dapat dilakukan secara acak.

2. Non volatile memory

Memory jenis ini kehilangan data yang dimuatnya pada saat tidak ada daya yang diberikan dan dikenal juga sebagai ROM (*Read Only Memory*). Sekali tulis dengan pola bit tertentu, tidak dapat lagi diubah. Beberapa jenis ROM dapat ditulis ulang. Penulisan ini dilakukan secara khusus sesuai dengan jenisnya.

3. ROM

ROM mengacu pada mask programmed ROM. Penulisan dilakukan melalui proses pembuatan IC. Setelah diproduksi, tidak dapat lagi dilakukan perubahan pola bit.

4. EPROM (*Erasable Programmable Read Only Memory*)

Penulisan dan penghapusan data dapat dilakukan oleh pengguna. Penghapusan dilakukan dengan menggunakan cahaya ultraviolet (UV) melalui jendela yang disediakan dikemasannya.

5. EEPROM (*Electrical Erasable Programmable Read-Only memory*)

Penghapusan EEPROM dilakukan dengan menggunakan tegangan listrik dengan besaran tertentu. Besaran tegangan ini tergantung pada produsennya.

Sistem minimal mikroprocessor dan mikrokontroler harus memiliki ROM. Pada saat dihidupkan, perintah yang disimpan pada ROM akan dijalankan. Perintah ini melakukan fungsi-fungsi dasar, termasuk akses RAM. Akses data ROM dilakukan secara random seperti pada ROM. Pada sistem mikrokontroler, terdapat memori *on-chip* dengan alamat yang sudah ditentukan.

### **2.6.5. Input / Output (I/O)**

Mikrokontroler mempunyai banyak pin yang dapat digunakan sebagai input/output (I/O). Setiap pin dapat mengeluarkan dan menerima masukan analog atau digital. Pin-pin I/O digital hanya dapat menerima masukan berupa tegangan pada level logika 0 atau 1. Pada setiap kesempatan pin I/O hanya dapat melakukan satu fungsi, sebagai masukan atau keluaran saja.

### **2.6.6. Interupsi**

Interupsi merupakan kondisi logika yang mewakili permintaan dengan prioritas khusus. Interupsi dapat dikenali secara perangkat keras atau perangkat lunak. Interupsi perangkat keras memiliki skala prioritas dan cara konfigurasi yang tetap. Interupsi perangkat lunak merupakan multipleks dari interupsi perangkat keras yang dapat diprogram ulang skala prioritas dan konfigurasinya. Alamat perintah yang dituju untuk permintaan yang berbeda disebut vektor interupsi.

Skala prioritas interupsi perangkat lunak memiliki indeks dalam bentuk bilangan. Bilangan yang indeksnya lebih rendah menyatakan tingkat prioritas yang lebih tinggi.

Secara umum, mikrokontroler menempatkan vektor interupsi reset sebagai vektor interupsi dengan indeks paling rendah.

### 2.6.7. Clock sistem dan osilator

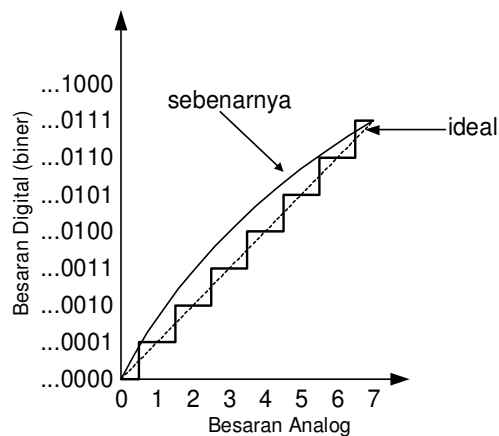
Mikrokontroler membutuhkan pulsa yang kontinu dengan frekuensi tertentu. Pulsa akan memicu mikrokontroler menjalankan perintah dan fungsi-fungsi yang telah diberikan. Referensi semua kerja yang dilakukan mikrokontroler berdasarkan pada clock sistem. Beberapa sumber clock yang dapat digunakan adalah rangkaian osilator RC, quartz crystal dan ceramic resonator.

### 2.6.8. Pengubah analog ke digital

Pengubah besaran analog menjadi besaran digital biasa disebut dengan ADC (*Analog digital converter*). Perubahan besaran dilakukan supaya besaran analog dapat diproses secara digital tanpa merubah kesetaraan nilai besaran analognya. Parameter yang menentukan mutu dari sebuah konverter ADC adalah :

1. Kesalahan kuantitasi
2. Ketidak-linieran
3. Kode tidak lengkap (hilang)
4. Waktu konversi

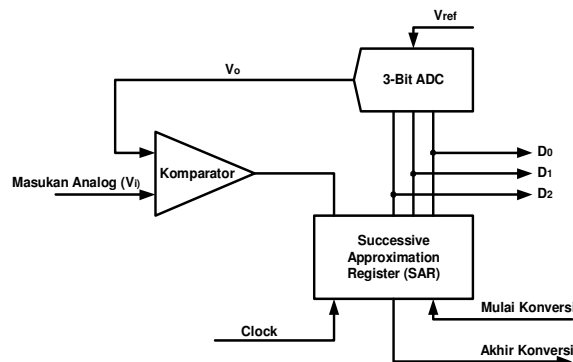
Karakteristik konversi besaran analog ke besaran digital dapat dilihat pada gambar 2.15 dibawah ini.



Gambar 2.15. Karakteristik konversi analog ke digital

Terdapat 3 tipe dari ADC yang dapat dijumpai, yakni :

1. *Parallel*, proses konversi ADC dengan metode paralel ini adalah dengan menggunakan fungsi komparator untuk setiap tingkat tegangan pembanding. Metoda ini membutuh banyak komparator untuk dapat menghasikan resolusi yang baik.
2. *Counting Type*, Metoda ini menggunakan pencacah yang terus naik yang kemudian dikonversikan menjadi digital ke analog (DAC). Tegangan keluaran DAC dibandingkan dengan tegangan analog masukan. Jika keluaran DAC lebih besar dari tegangan masukan, maka pencacah berhenti. Nilai hasil cacahan merupakan nilai digital dari tegangan analog masukan.
3. *Successive Approximation*. Tipe successive approximation (pendekatan berturut-turut) banyak dipakai karena mendukung resolusi tinggi dan kecepatan konversi yang tinggi. Pada gambar 2.16 terlihat blok diagram metoda successive approximation.



Gambar 2.16. ADC tipe Successive approximation

Pada dasarnya metoda ini terbagi dalam 4 bagian, yaitu *Control logic*, *Digital to Analog Converter (DAC)*, *Successive Approximation Register (SAR)* dan Komparator.

Konversi dimulai dengan mengaktifkan pin start pada blok kontrol logik. Kemudian komparator akan membandingkan tegangan analog  $V_i$  yang tidak diketahui nilainya dibandingkan dengan tegangan referensi dimulai dari  $V_{ref}/2$ . Rangkaian logika diletakkan berurutan dan dites setiap bitnya, dimulai dari *most significant bit (MSB)*. Setelah semua bit pada kondisi low maka operasi pertama  $a_0=1$  dan diuji  $V_f = V_{ref} \cdot 2^{-1}$  dibandingkan dengan  $V_i$  pada komparator. Jika  $V_i$  lebih besar dari  $V_o$  maka didapat



logika high dan sebaliknya jika lebih kecil didapat logika low. Proses tersebut berlanjut sebanyak jumlah bit dari ADC tersebut.

Untuk mendapatkan nilai desimal dari sebuah tegangan analog dipakai rumus :

$$N(\text{desimal}) = \frac{V_i}{V_{ref}} \times 2^n$$

dimana :

- $V_i$  = tegangan input
- $V_{ref}$  = tegangan referensi
- $n$  = jumlah bit
- $N$  = nilai desimal

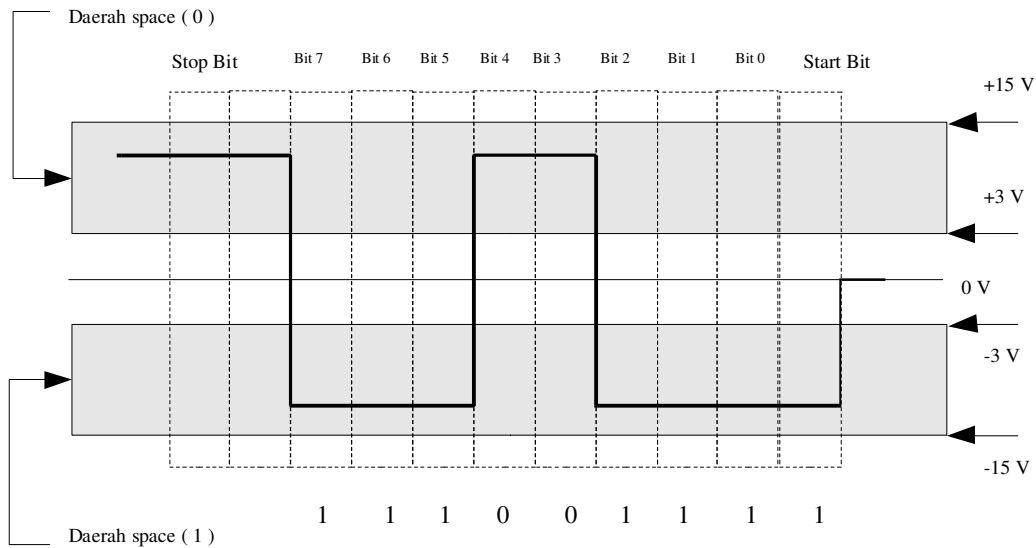
### 2.6.9. Komunikasi data serial RS-232

Komunikasi data serial bagian dari lapisan fisik dari model OSI. Bit-bit dikirimkan satu bit untuk setiap waktu. Bit 1 disebut sebagai *mark* dan bit 0 disebut sebagai *space*. Kecepatan transmisi data dinyatakan dalam bit perdetik (bps).

Transmisi dapat dilakukan dengan menggunakan jalur yang sama secara bergantian yang disebut dengan *half duplex*. Transmisi dapat juga dilakukan dengan memisahkan jalur pengiriman dan penerimaan yang disebut dengan *full duplex*.

Komunikasi data serial dapat dilakukan secara sinkron dan asinkron. Secara sinkron, data dikirim sesuai waktu pengiriman yang ditetapkan, sehingga waktu pengiriman harus tepat dan sama antara kedua sisi. Secara sinkron, data dapat dikirim pada waktu yang acak dan waktu pengiriman yang dinegosiasikan pada awal pengiriman data.

Salah satu protokol komunikasi serial asinkron yang banyak digunakan adalah RS232-C (V2.4) dari EIA( *Electrical industry Association*). EIA RS-232-C merekomendasikan mark dikirim pada level tegangan antara -3 hingga -15 V dan space pada level tegangan +3 hingga +15 V. Jarak antara DTE dan DCE sejauh 50 feet (15 meter) dan laju bit yang digunakan 9600 bps.



Gambar 2.17 Sinyal transmisi rs232c

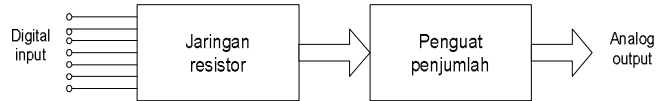
Jumlah bit dalam 1 byte harus disepakati sebelum transmisi. 7 bit dan 8 bit merupakan jumlah bit yang lazim dipergunakan dalam komunikasi data serial. Pada gambar 2.17, jumlah bit untuk setiap byte yang ditransmisikan adalah 8-bit.

Start dan stop bit merupakan bit yang melakukan kontrol transmisi dan membungkus setiap byte data yang akan ditransmisikan. Start bit memberitahu DTE akan ada data yang dikirim. Stop bit menyatakan akhir dari transmisi dan waktu DTE harus menunggu hingga byte selanjutnya dikirimkan.

Bit paritas merupakan mekanisme pemeriksaan sederhana terhadap kebenaran setiap byte yang telah dikirimkan. Bit paritas dapat berupa genap dan ganjil. Bit paritas ditentukan dengan menjumlahkan nilai *mark* (1) pada setiap byte data dan jenisnya disepakati diawal transmisi. Jika paritas disepakati genap, maka bit paritas pada gambar diatas dinyatakan dengan 0, karena jumlah 1 pada byte adalah genap. Jika paritas disepakati ganjil, maka paritas haruslah 1, karena jumlah 1 pada byte adalah genap.

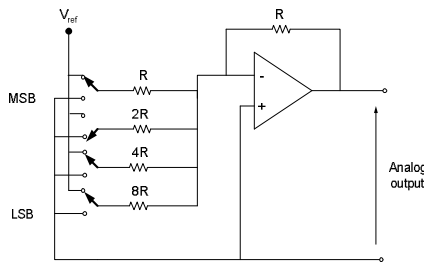
## 2.7. Pengubah digital ke analog

Pada konverter digital dan analog tersusun dari rangkaian resistor yang mengubah setiap bit sinyal masukan menjadi arus yang sesuai. Jumlah masing-masing arus tersebut mengalir ke resistor yang mengubahnya menjadi tegangan yang proporsional. Gambar 2.18 menunjukkan elemen dasar konverter digital ke analog.



Gambar 2.18. Elemen dasar DAC

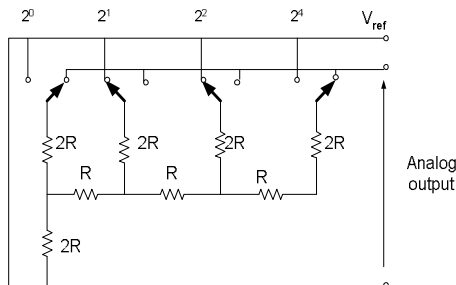
Rangkaian penjumlahan op-amp dapat digunakan untuk menyusun suatu konverter D/A dengan memakai sejumlah hambatan masukan yang diberi bobot dalam deret biner. Pada gambar 2.19 dapat dilihat.  $V_{ref}$  adalah tegangan acuan(referensi) dan resistansi yang digunakan adalah hambatan yang presisi agar mendapatkan nilai arus masukan yang tepat.



Gambar 2.19. Rangkaian konverter dengan bobot biner

Kekurangan dari konverter dengan menggunakan hambatan berbobot adalah dengan meningkatnya jumlah bit, jangkauan resistansi yang dihasilkan menjadi besar dalam waktu singkat dan toleransi kesalahan menjadi besar seiring dengan naiknya resolusi konverter.

Salah satu cara untuk mengatasi masalah yang timbul akibat bobot biner adalah dengan menggunakan rangkaian tangga (*ladder*). Contoh dari rangkaian tangga R-2R dapat dilihat pada gambar 2.20. Rangkaian ini hanya memerlukan dua nilai hambatan yaitu R dan 2R. Rangkaian R - 2R adalah nilai impedansi pada titik yang berbeda pada rangkaian akan membentuk susunan paralel dan direduksi menjadi hambatan R .



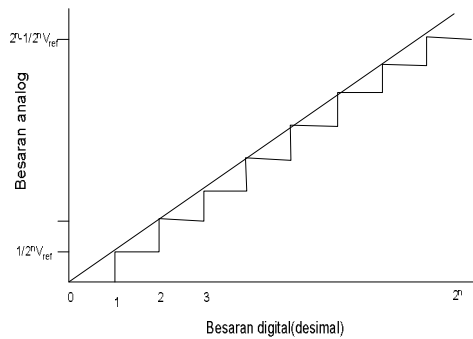
Gambar 2.20. Rangkaian konverter tangga

Skema dasar rangkaian dac dengan menggunakan hubungan R-2R dapat dilihat pada gambar diatas. Apabila kita memindahkan posisi saklar kita akan mengubah pembagian arus pada tiap-tiap simpul. Untuk mendapatkan nilai arus dapat dihitung dengan

$$I_{out} = (D_{n-1} + 2^{-1} D_{n-2} + \dots + 2^{-1-n} D_0) \frac{I_{ref}}{2} \quad (2.57)$$

Waktu ayun adalah waktu dari saat perubahan sinyal masukan waktu digital hingga konverter mencapai nilai 1/2 dari nilai analognya. Konverter D/A yang ada sekarang mempunyai waktu ayun sekitar 100 nanodetik hingga hingga 5 mikrodetik pada penguraian dari 8 bit hingga 14 bit.

Karakteristik konversi besaran digital ke besaran analog dapat dilihat pada gambar 2.21 dibawah ini.



Gambar 2.21. Karakteristik konversi digital ke analog

Secara umum untuk menghitung tegangan keluaran maksimum dari dac dapat dihitung dengan :

$$V_{out} = \frac{(2^n - 1)}{2^n} \times V_{ref}$$

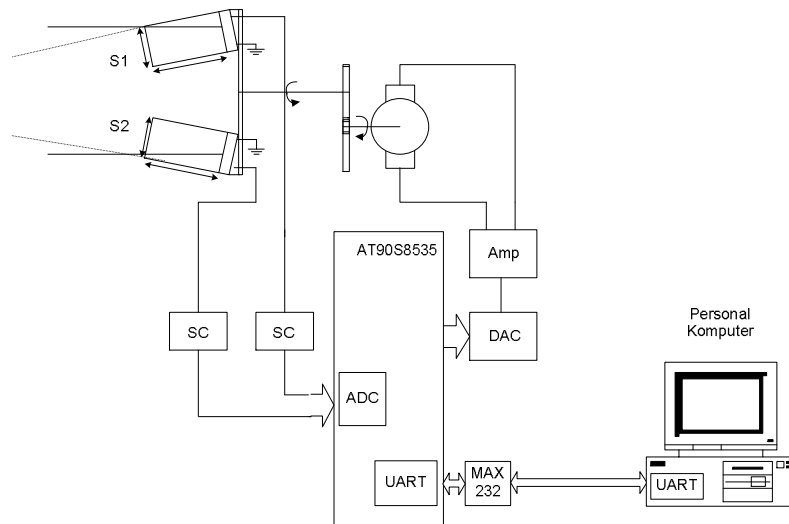
dimana :  $V_{out}$  = tegangan input  
 $V_{ref}$  = tegangan referensi  
 $n$  = jumlah bit

## Bab 3

### PERANCANGAN SISTEM BESERTA SUBSISTEM

#### 3.1. Gambaran umum sistem

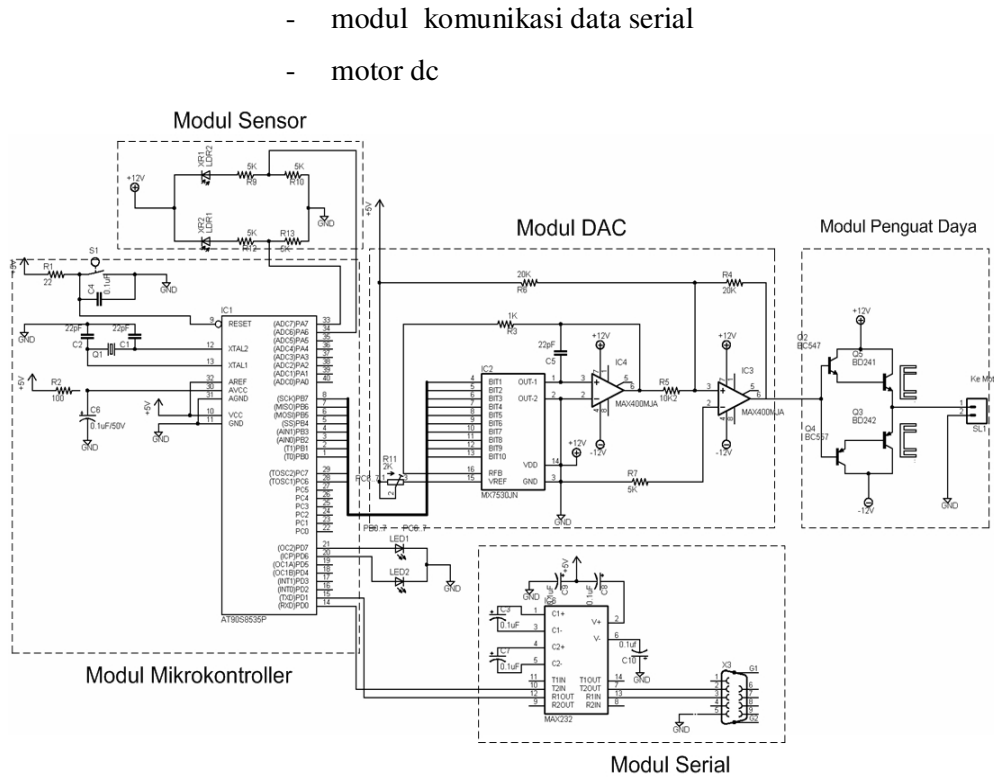
Energi surya dapat digunakan sebagai sumber energi alternatif. Dengan penggunaan kolektor surya diharapkan dapat menangkap energi cahaya matahari secara maksimal. Posisi kolektor surya terhadap arah cahaya matahari sangat mempengaruhi banyaknya energi surya yang akan dimanfaatkan. Pada penelitian ini dirancang sebuah sistem penggerak kolektor energi surya dengan mikrokontroler. Sistem ini terdiri dari beberapa elemen yang hubungannya dapat dilihat pada gambar 3.1



Gambar 3.1. Rancangan sistem

Sistem ini terdiri dari beberapa elemen yaitu

- modul mikrokontroler
- modul sensor dan *signal conditioning*
- modul mikrokontroler AT 90S8535
- modul DAC
- modul penguat arus motor



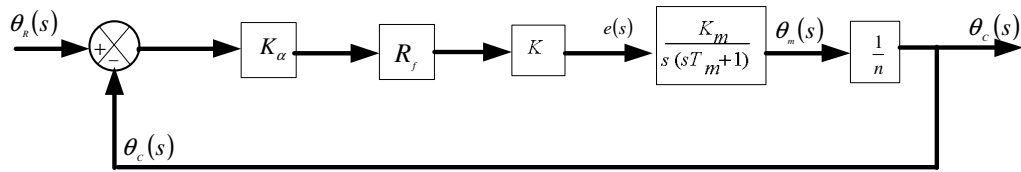
Gambar 3.2. Skema rangkaian penggerak kolektor surya

Prinsip kerja sistem secara umum adalah :

- LDR1 dan LDR2 pada gambar 3.2, mengukur intensitas cahaya matahari. Sensor mengkonversikan intensitas cahaya yang mengenainya menjadi besaran listrik. Data akan dibaca oleh ADC pada pin 33 dan 34 mikrokontoller AT90S8535.
- Mikrokontroler diprogram untuk mengolah data pembacaan sensor. Secara umum, program akan membandingkan nilai yang didapatkan oleh kedua sensor, dan menganalisa data yang didapat. Hasil pengolahan data akan menentukan besar tegangan yang dihasilkan oleh modul dac. Penguat daya akan menyuplai daya yang dibutuhkan motor.
- Sistem ini dirancang dengan 2 metode operasi. Sistem dapat dijalankan dengan dan tanpa menggunakan komputer. Untuk sistem yang terhubung dengan PC, komunikasi komputer dengan mikrokontroler menggunakan komunikasi data *half duplex* dengan antarmuka RS232. PC dapat

menjalankan peralatan penggerak dan membaca setiap data yang dikirimkan oleh mikrokontroler untuk divisualisasikan pada komputer.

Sistem kontrol ini merupakan substitusi dari sistem kontrol posisi yaitu pengontrolan posisi penggerak kolektor surya. Sistem pengontrolan ini menggunakan 1 titik gangguan dengan 1 titik kontrol. Dengan mensubstitusi kedua sistem kontrol tersebut maka di dapat sebuah blok diagram dengan 1 titik gangguan. Rancangan blok diagram sistem ini dapat dilihat pada gambar 3.3

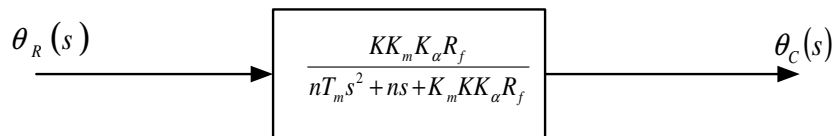


Gambar 3.3. Blok diagram sistem tanpa gangguan

Keterangan :

- $\theta_R(s)$  = Sudut setting antara sensor terhadap sumber cahaya
- $\theta_C(s)$  = Sudut yang terjadi antara sensor terhadap sumber cahaya
- $K_\alpha$  = konstanta hubungan beda tegangan sensor dan posisi sudut
- $R_f$  = rasio penguatan tegangan sensor
- $K_m$  = konstanta penguatan motor
- $T_m$  = konstanta penguatan motor
- $n$  = rasio gir motor penggerak

Dari blok diagram pengontrolan diatas dapat disederhanakan menjadi :



Gambar 3.4 Blok diagram sistem yang disederhanakan

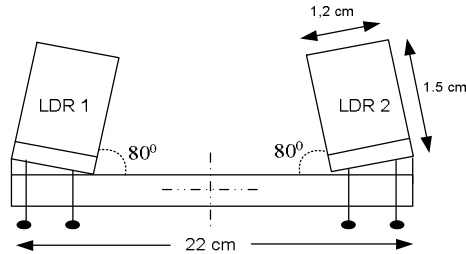
Dengan nilai fungsi  $K_\alpha(t)$  adalah merupakan fungsi tegangan terhadap sudut, yang memiliki karakteristik sebagai berikut

$$= 0 \quad \text{untuk } \alpha < -25^0 \text{ atau } \alpha > 25^0$$

$$K_{\alpha}(t) = K_{\alpha}\alpha(t) \text{ untuk } 25^{\circ} < \alpha < 25^{\circ}$$

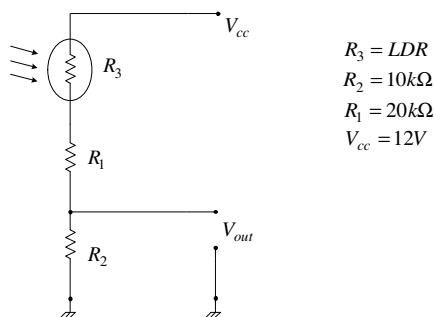
### 3.3. Perancangan modul sensor

Sensor yang digunakan yaitu 2 buah LDR yang identik. Rancangan sensor yang digunakan dapat dilihat pada gambar 3.5.



Gambar 3.5 Rancangan sensor

Pada sistem ini digunakan modul adc pada mikrokontroler dengan tegangan masukan maksimum +5 VDC. Untuk membatasi tegangan masuk ADC, digunakan pembagian tegangan seperti pada gambar 3.6.



Gambar 3.6 Rangkaian pembagi tegangan pada sensor

$V_{out}$  merupakan tegangan pada  $R_2$ .  $R_3$  adalah (LDR) dengan tahanan saat terang sekitar  $2\text{ k}\Omega$  dan lebih kurang  $6\text{ M}\Omega$  pada keadaan gelap. Untuk mendapatkan  $V_{out} = 4$  volt dengan  $V_{cc} = 12$  volt dengan  $R_2 = 20\text{ k}\Omega$  disusun seri dengan  $R_1$ , sehingga didapat nilai  $R_1$  adalah :

$$V_{out} = \frac{R_2}{R_1 + R_3 + R_2} V_{cc}$$



$$4 = \frac{R_2}{2.10^4 + 2000 + R_2} 12$$

$$R_2 = 11.000 \Omega$$

Karena tahanan sebesar  $11 \text{ k}\Omega$  tidak tersedia, maka  $R_2$  dapat diganti dengan tahanan sebesar  $10 \text{ k}\Omega$  sehingga pada saat gelap tegangan keluaran maksimum adalah  $V_{out} = 3,75 \text{ Volt}$ . Dengan persamaan diatas dapat juga dihitung besarnya penguatan tegangan yang terjadi adalah

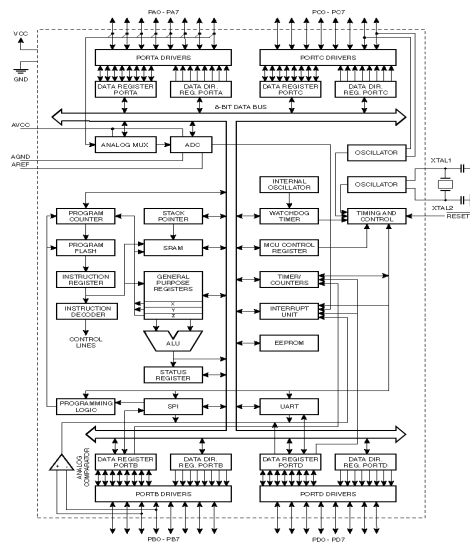
$$\frac{V_{R2}}{V_{R3}} = \frac{10000}{2000} = 5$$

Sehingga didapat nilai konstanta  $R_f$  pada sistem adalah  $R_f = 5$ .

### 3.2. Pemanfaatan mikrokontroler AT90S8535<sup>[4]</sup>

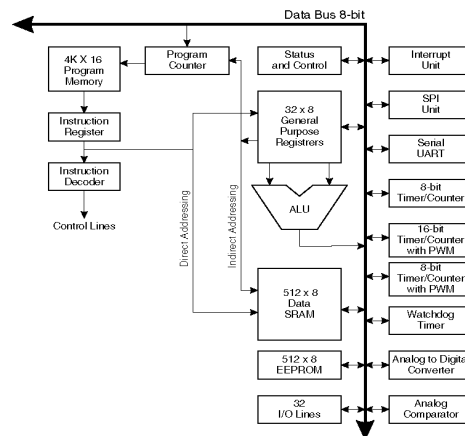
AT90S8535 adalah 8-bit CMOS mikrokontroler yang berbasis arsitektur RISC, dan dilengkapi dengan 32 register yang bekerja secara umum (*General-Purpose Register*). Register tersebut terhubung langsung ke ALU, dan beberapa diantaranya dapat digunakan dua buah register secara langsung dan bebas serta dapat diakses dengan satu perintah dalam satu kali periode clock.

Pada gambar 3.7 menunjukkan blok diagram mikrokontroler AT90S8535. Dalam gambar tersebut terlihat fasilitas-fasilitas yang terdapat pada mikrokontroler tersebut.



Gambar 3.7. Blok diagram mikrokontroler AT90S8535

- Mikrokontroler AT90S8535 ini juga dilengkapi dengan 8 kilobyte *flash* memori yang dapat diprogram, 512 byte *static memory* (SRAM), 512 byte EEPROM, 32 jalur I/O yang dapat berganti fungsi, *real-time clock* (RTC), 8 bit dan 16 bit timer/counter, analog komparator, UART, 8 kanal 10 bit ADC, *master/slave* SPI (*serial peripheral interface*), internal dan external interupsi, *programmable watchdog timer* dengan internal osilator, dan 2 *software-selectable power-saving modes*.

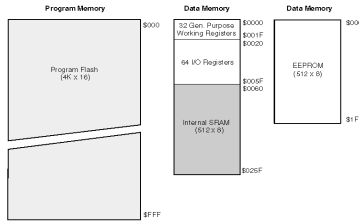


Gambar 3.8 Arsitektur RISC pada AT90S8535

Pada gambar 3.8, terlihat 32 x 8-bit register yang terhubung langsung ke ALU sehingga dapat diakses dalam waktu satu clock. Keadaan ini menunjukkan bahwa dua data dari register dieksekusi kemudian disimpan kembali ke register dalam 1 periode clock.

Teknologi arsitektur *Harvard* digunakan pada mikrokontroler AT90S8535, dengan memisahkan memori dan bus dari program data. Input/Output (I/O) pada AT90S8535 berjumlah 64 alamat, yang berfungsi untuk kontrol register, timer/counter, konverter analog-digital dan fungsi-fungsi I/O lainnya.

Memori pada arsitektur AT90S8535, terbagi atas flash memori, EEPROM, SRAM serta register dan I/O. Gambar 3.9 menunjukkan peta memori pada mikrokontroler AT90S8535 .

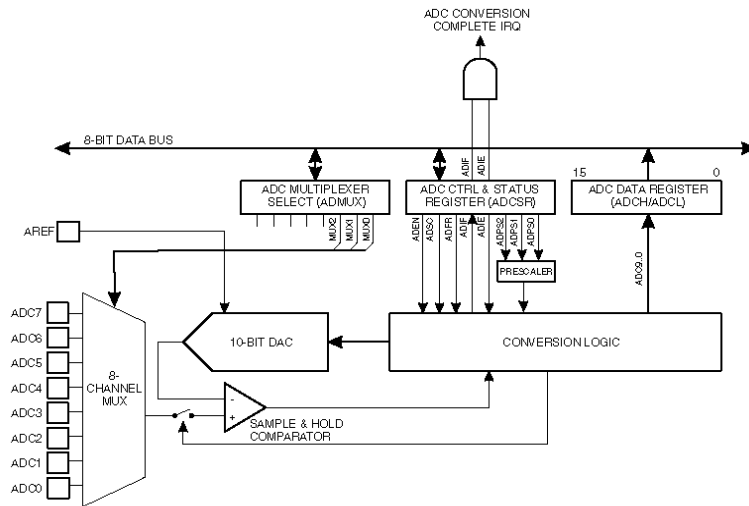


Gambar 3.9. Peta memori AT90S8535

SRAM pada mikrokontroler AT90S8535 terdiri dari 32 *General-Purpose Register*, register untuk input/output (I/O) dan internal RAM.

### 3.2.1. Pemanfaatan ADC pada AT90S8535<sup>[4]</sup>

AT90S8535 memiliki ADC (*Analog Digital Converter*) dengan resolusi 10 bit dengan waktu konversi 65–260  $\mu$ s dengan akurasi kesalahan  $\frac{1}{2}$  LSB. ADC pada mikrokontroler AT90S8535 terdiri dari 3 register I/O, tegangan referensi dan sebuah multiplex 8 kanal. Gambar 3.10 menunjukkan blok skema ADC pada mikrokontroler AT90S8535.



Gambar 3.10. Blok diagram ADC AT90S8535

Pada register kontrol dan status terdapat jalur yang menghubungkan dengan vektor interupsi. ADC pada mikrokontroler ini menggunakan metode *successive approximation*. Register input/output tersebut antara lain :

- **ADMUX (ADC multiplexer select register)**, Register ini berguna untuk memilih kanal adc yang digunakan. Alamat register tersebut adalah \$07 (\$27). Pemilihan

dilakukan dengan mengisi register ADMUX tersebut dengan data, dengan ketentuan :

- Bit 7,...3, diabaikan.
- Bit 2,...0, jalur ADC yang akan digunakan.

Untuk pemilihan jalur ADC dapat dilihat pada tabel 3.1.

Tabel 3.1 : Pemilihan kanal ADC

MUX	Channel
000	ADC0
001	ADC1
010	ADC2
011	ADC3
100	ADC4
101	ADC5
110	ADC6
111	ADC7

- ADCSR (ADC Control and Status Register)

Register ini berfungsi untuk mengatur fungsi ADC (ADC enable), memulai konversi, akhir konversi, memilih mode operasi ADC, mengaktifkan

Bit	7	6	5	4	3	2	1	0
Alamat \$06 (\$26)	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0

interupsi, dan memilih frekuensi ADC.

- Bit 7 – ADEN, bit yang harus di set 1 bila adc diaktifkan.
- Bit 6 – ADSC, adc memulai konversi jika bit ini diset 1.
- Bit 5 – ADFR, penyeleksian mode *free running* adc bila bit ini diset 1.
- Bit 4 – ADIF, bendera interupsi adc, bit ini akan bernilai 1 jika adc telah siap melakukan konversi.
- Bit 3 – ADIE, interupsi adc diaktifkan, bila bit ini diset 1 maka ketika adc telah siap melakukan interupsi maka akan melakukan internal interupsi.

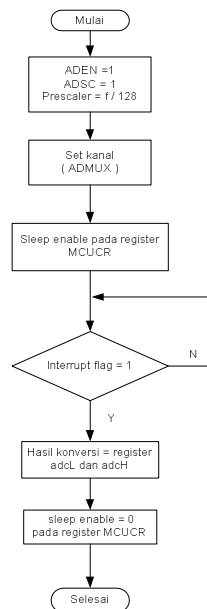
- Bit 2,...0 – ADPS2,...ADPS0, bit yang digunakan untuk pemilihan *prescaler* ADC. Nilai ini menentukan clock yang digunakan oleh ADC. Tabel 3.2 memperlihatkan nilai *prescaler*. Clock yang akan digunakan ADC dihitung dengan persamaan

$$\text{Clock} = \frac{\text{Clock}_{\mu\text{C}}}{\text{prescaler}}$$

- ADCL and ADCH (ADC Data Register )  
Jika konversi telah selesai maka data hasil konversi akan tersimpan pada register ini. ADCL merupakan 8 bit data bawah dan ADCH terdapat 2 bit data atas data dari hasil konversi ADC.

Untuk mengaktifkan ADC ada 3 *power supply* analog yang harus dipenuhi :

- AVCC merupakan suplay analog yang bernilai  $VCC \pm 0.3V$
- AGND merupakan analog ground yang dapat dihubungkan ke *ground*
- AREF merupakan tegangan referensi external yang bernilai  $2.0- AVCC$ .



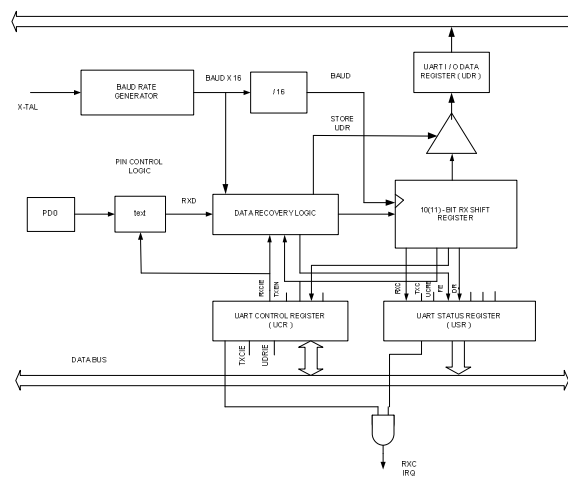
Gambar 3.11 Diagram alir konversi adc dengan interupsi

Pada rancangan ini digunakan interupsi dari ADC. ADC akan memberikan interupsi kepada mikrokontroler jika proses konversi sudah selesai. Selama proses konversi berlangsung mikrokontroler akan *standby* sampai ada interupsi untuk membaca

data hasil pada register adcL dan adcH. Rancangan ini menggunakan 2 kanal adc yaitu kanal 0 dan kanal 1 untuk 2 buah sensor dengan tegangan referensi +5 VDC.

### 3.2.2. Pemanfaatan UART pada AT90S8535<sup>[4]</sup>

AT90S8535 memiliki UART (*Universal Asynchronous Receiver and Transmitter*) yang mengatur pengiriman dan penerimaan karakter secara serial tidak sinkron (*Asynchronous serial transmission*). UART mengizinkan transmisi *full duplex*, pengaturan *baudrate*, panjang data 8 atau 9 bit, filter derau dan fasilitas lainnya. Blok diagram UART dalam AT90S8535 dapat dilihat pada gambar 3.12. Desain UART terdiri dari 3 bagian, yaitu pengirim, penerima dan kontrol.



Gambar 3.12. Blok diagram UART pada AT90S8535

Proses pengiriman dan penerimaan karakter melalui jalur serial diatur dengan menggunakan register-register kontrol UART yang terdiri dari :

- UDR (UART I/O Data Register)  
Menyimpan karakter yang akan dikirim dan diterima
- USR ( UART Status Register )  
Menyimpan status pengiriman, penerimaan, kondisi register geser dan kesalahan yang terjadi.
- UCR ( UART Control Register )

Mengatur aktivasi parameter komunikasi data serial, meliputi interupsi UART, blok pengirim, blok penerima, jumlah bit karakter dan jumlah bit pengiriman atau penerimaan

- UBRR ( UART Baud Rate Register )

Mengatur bit rate yang digunakan dalam komunikasi serial

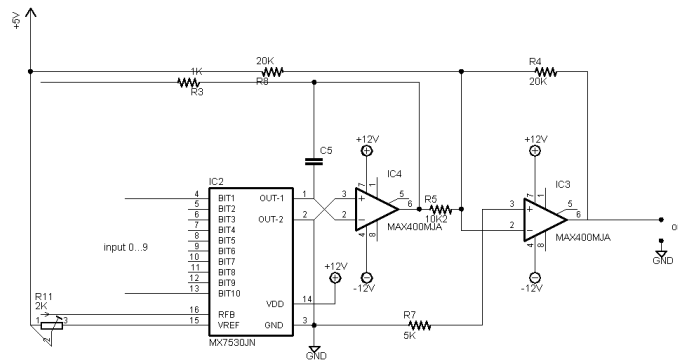
Fasilitas UART harus melalui proses inisialisasi dengan melakukan penyettingan *baudrate*, aktivasi blok pengirim, blok penerima. Penyettingan *baudrate* dilakukan dengan memberikan nilai tertentu kepada UBRR. Nilai setting *baudrate* dengan clock 8 MHz dapat dilihat pada tabel 3.2. Aktivasi parameter komunikasi data serial bit dapat dilakukan dengan mengaktifkan bit-bit terkait pada UCR.

Tabel 3.2 Tabel nilai setting UBRR untuk clock 8 Mhz

Baudrate	UBRR	%
2400	207	0.2
4800	103	0.2
9600	51	0.2
14400	34	0.8
19200	25	0.2
28800	16	2.1
38400	12	0.2
57600	8	3.7
76800	6	7.5
115200	3	7.8

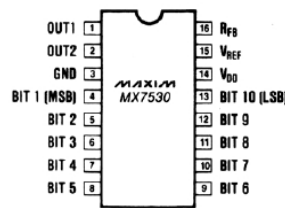
### 3.4. Perancangan modul DAC dan hubungannya dengan mikrokontroler

Modul DAC yang digunakan dapat mengeluarkan tegangan  $\pm 11$  VDC dengan waktu ayun 500 ns. Dalam perancangan ini digunakan batas tegangan keluaran sebesar  $\pm 5$  VDC sesuai tegangan kerja pada motor yang digunakan. Untuk mengubah batas tegangan tersebut dilakukan dengan mengatur tegangan referensi ( $V_{ref}$ ) pada chip DAC. Modul DAC terdiri dari sebuah DAC tipe *MX7530* dan penguat tipe *MAX400* yang keduanya keluaran *Maxim*. Skema rangkaian modul dac dapat dilihat pada gambar 3.13.



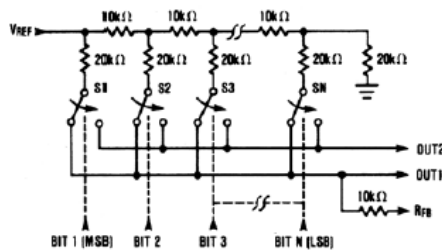
Gambar 3.13 Skema rangkaian modul dac

MX 7530 adalah salah satu tipe DAC tipe pengali cmos dengan resolusi masukan 10 bit. Chip ini bekerja pada tegangan operasi +5 sampai dengan -15 VDC dengan konsumsi daya 450 mW.



Gambar 3.14. Konfigurasi pin-pin MX7530

Chip ini tersusun dari gabungan resistor film tipis dengan hubungan R-2R yang memakai teknik penyambungan dengan laser, serta saklar arus CMOS. Pada gambar 3.15 terlihat elemen-elemen penyusun dari chip MX7530.



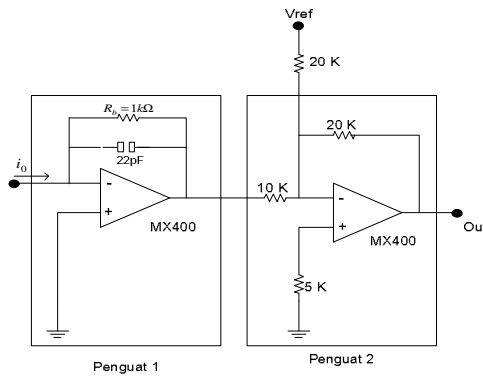
Gambar 3.15. Konstruksi MX7530

R<sub>1</sub> digunakan untuk mengubah tegangan referensi dan C<sub>1</sub> adalah kapasitor kompensasi yang mencegah terjadinya perubahan kutub akibat dari kapasitansi dan



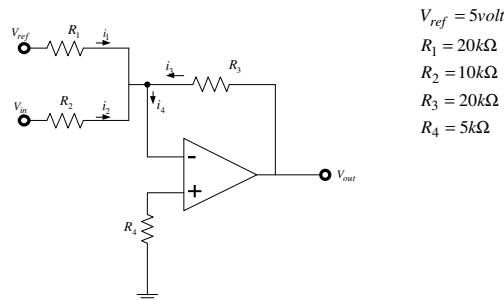
tahanan dalam keluaran ketika menggunakan penguat dengan waktu pensaklaran yang kecil.

Pada modul ini digunakan 2 buah op-amp sebagai integrator dan penguat penjumlah pembalik (*inverting*). Op-amp yang digunakan adalah tipe MX400 keluaran *Maxim*. Penguat tipe ini karena mempunyai batasan tegangan yang lebar dari  $\pm 3$  VDC sampai dengan  $\pm 18$  VDC tanpa perlu pengaturan tegangan offset.



Gambar 3.16. Konfigurasi op-amp pada modul dac

Pada gambar 3.16 terlihat hubungan dari kedua penguat. Untuk penguat pertama akan dihasilkan tegangan sebesar  $-(i_0 \times R_b)$ . Sedangkan penguat kedua adalah penguat penjumlah, seperti terlihat pada gambar 3.17. dibawah ini



Gambar 3.17. Rangkaian penguat penjumlah pada modul

Dimana jumlah arus dititik a = 0, sehingga  $i_1 + i_2 + i_3 + i_4 = 0$  maka didapatkan nilai penguatan tegangan sebesar :

$$V_{out} = -\left[\frac{R_3}{R_2}\right]V_{ref} - \left[\frac{R_3}{R_1}\right]V_{in}$$

$$V_{out} = -\left[\frac{2000}{2000}\right]V_{ref} + \left[\frac{2000}{1000}\right]xV_{in}$$

sehingga

$$V_{out} = -V_{ref} + 2V_{in}$$

dimana

$$V_{in} = -i_{out}xR_{fb}$$

maka

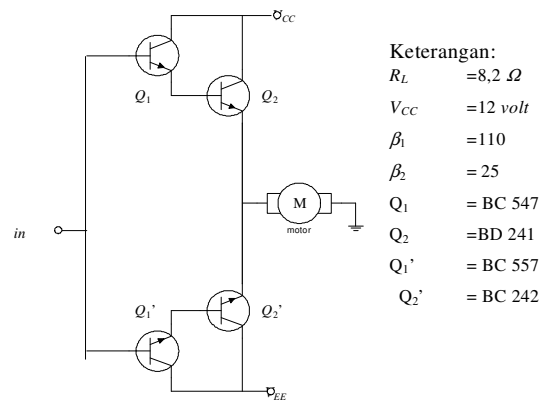
$$V_{out} = -V_{ref} + (2 \times 1000 \times i_0)$$

$$V_{out} = -V_{ref} + 2000i_0$$

Nilai  $i_0$  dapat dihitung dengan menggunakan persamaan (2.57)

### 3.5. Perancangan modul penguat arus penggerak motor DC

Pada rancangan ini motor yang digunakan adalah motor dc pengaturan arus jangkar dengan medan magnet permanen. Rancangan penguat arus dapat dilihat pada gambar 3.18.



Gambar 3.18. Rancangan penguat arus motor

Besar arus keluar pada transisitor kedua dapat dihitung dengan menggunakan cara

$$\begin{aligned} \beta_2 &= 25 \\ I_O &= I_{C2} + I_{B2} \\ &= 26I_{B2} \end{aligned}$$

$$\beta_1 = 110$$

$$I_{B2} = I_{C1} + I_{B1}$$

$$= 111I_{B1} \quad \text{analisa dc, } I_{B1} = I_{in}$$

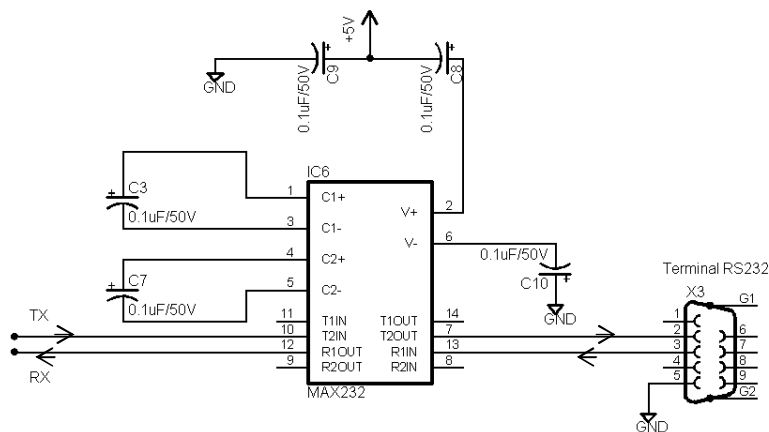
$$= 111I_{in}$$

$$I_O = 2886I_{in}$$

maka penguatan arus adalah 2886

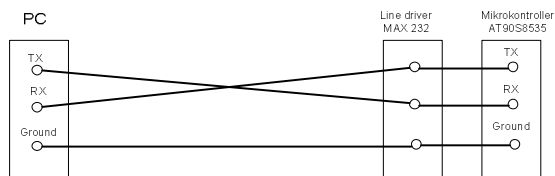
### 3.6. Perancangan modul komunikasi serial

Modul komunikasi serial digunakan supaya sinyal serial yang dikeluarkan oleh UART mikrokontroler dapat ditransmisikan sesuai dengan standarisasi *EIA-232*. Dengan modul ini sistem dapat berkomunikasi dengan komputer dengan komunikasi data secara serial.



Gambar 3.19 Skema rangkaian modul serial

Rancangan modul serial secara keseluruhan dapat dilihat pada gambar 3.19. Dalam komunikasi serial, jalur pengirim dan penerima kedua sisi komunikasi menyilang. Dalam hal ini, jalur pengirim pada UART mikrokontroler dihubungkan dengan jalur penerima PC. Jalur penerima mikrokontroler dihubungkan dengan jalur pengirim PC. Dalam perancangan ini, hanya digunakan komunikasi serial tiga jalur yang terdiri dari jalur tx, rx dan ground. Lebih jelasnya hubungan kabel ini diperlihatkan oleh gambar 3.20.



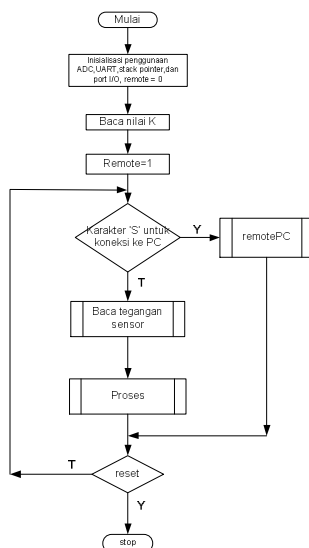
Gambar 3.20. Hubungan serial PC dan UART AT90S8535

### 3.7. Perancangan perangkat lunak sistem

Pada perancangan ini perangkat lunak akan mengatur penggunaan perangkat keras dan pemanfaatan komputer sebagai visualisasi hasil pengontrolan. Perangkat lunak atau program aplikasi yang digunakan secara umum terdiri atas perangkat lunak pada mikrokontroler dan pada PC.

#### 3.7.1. Perangkat lunak pada mikrokontroler

Pada mikrokontroler, perangkat lunak digunakan mengatur fungsi penggerak kolektor surya dan komunikasi data secara serial dengan PC. Diagram alir perangkat lunak pada mikrokontroler dapat dilihat pada gambar 3.21.

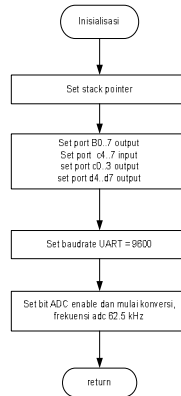


Gambar 3.21. Diagram alir perangkat lunak pada mikrokontroler

##### 3.7.1.1. Inisialisasi

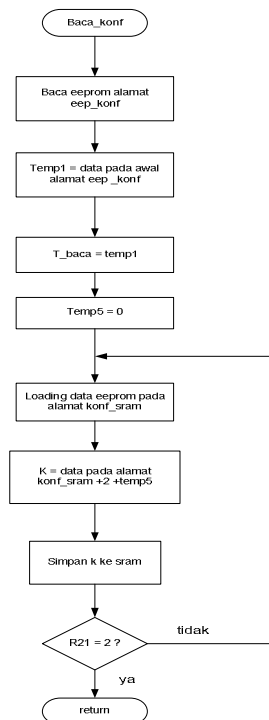
Pada proses ini diinisialisasikan fungsi-fungsi mikrokontroler yang akan digunakan. Proses meliputi inisialisasi register *stack pointer*, port b0, c4 sampai c7, d4 sampai d7 sebagai keluaran, port c4 sampai c7 sebagai masukan, mengaktifkan fungsi

uart pada baudrate 9600, dan mengaktifkan adc *on-chip* dengan frekuensi konversi 62.5 kHz.



### 3.7.1.2. Baca konfigurasi

Pada subrutin ini dilakukan pemindahan variabel konfigurasi yang disimpan di eeprom ke sram yang akan digunakan dalam kalkulasi perhitungan respon motor. Data konfigurasi yang disimpan pada eeprom adalah konstanta k.



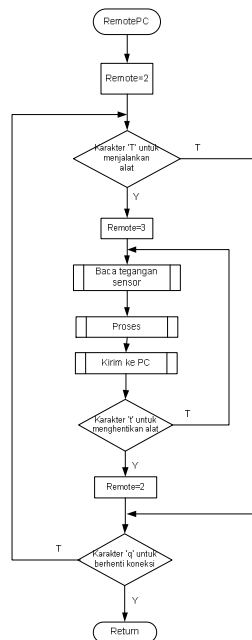
c. Hubungan ke PC

Pada subrutin ini perintah akan diberikan PC dan pengaturan kerja alat dilakukan PC.

Pada subrutin ini mikrokontroler akan mendeteksi perintah dari komputer yaitu :

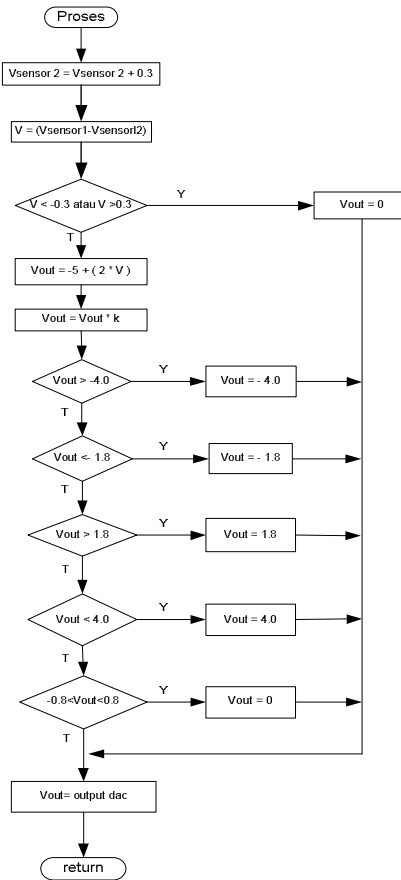
- Karakter 'S' untuk memulai hubungan dengan komputer. Ini akan mematikan proses yang dijalankan mikrokontroler
- karakter 'q' untuk mengakhiri hubungan dengan komputer
- karakter 'T' untuk menjalankan penggerak kolektor surya
- karakter 't' untuk menghentikan penggerak kolektor surya

Diagram alir proses ini dapat dilihat pada gambar dibawah ini.



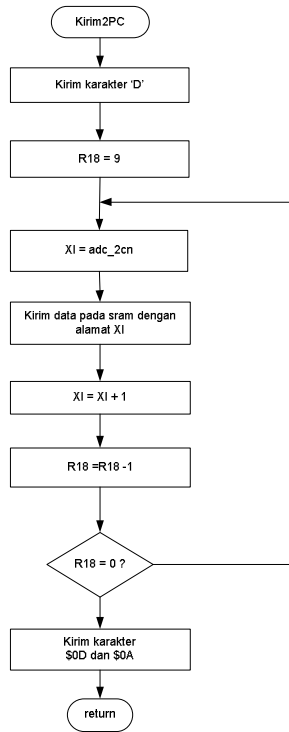
d. Pengolahan data

Pada proses ini dibaca nilai tegangan sensor, kemudian disimpan pada eeprom. Selisih tegangan kedua sensor digunakan sebagai penggerak motor dc. Hasil pengolahan merupakan data 8 bit port B dan 2 bit port C, yang dihubungkan pada modul dac.



e. Kirim data ke PC

Fungsi ini berguna untuk mengirimkan data hasil pembacaan ADC yang disimpan sram ke komputer. Fungsi ini mengirimkan karakter 'D' sebagai penanda awal data dan karakter 0x0D dan 0x0A sebagai tanda akhir pengiriman data.





## Bab 4

### PENGUJIAN SISTEM BESERTA SUBSISTEM

#### 4.1 Pengantar Umum

Untuk mengetahui kinerja dan karakteristik rancangan ini maka dilakukan pengujian-pengujian untuk mendapatkan karakteristik sistem dan subsistem pendukungnya. Adapun pengujian-pengujian yang dilakukan sebagai berikut :

1. Pengujian modul sensor
2. Pengujian karakteristik motor
3. Pengujian adc pada mikrokontroler
4. Pengujian modul dac
5. Pengujian modul komunikasi data serial
6. Pengujian sistem

#### 4.2. Pengujian dan analisa karakteristik sensor

##### 1. Tujuan

Tujuan pengujian ini adalah untuk mengetahui karakteristik sensor LDR dan rasio gir motor.

##### 2. Alat dan bahan yang digunakan

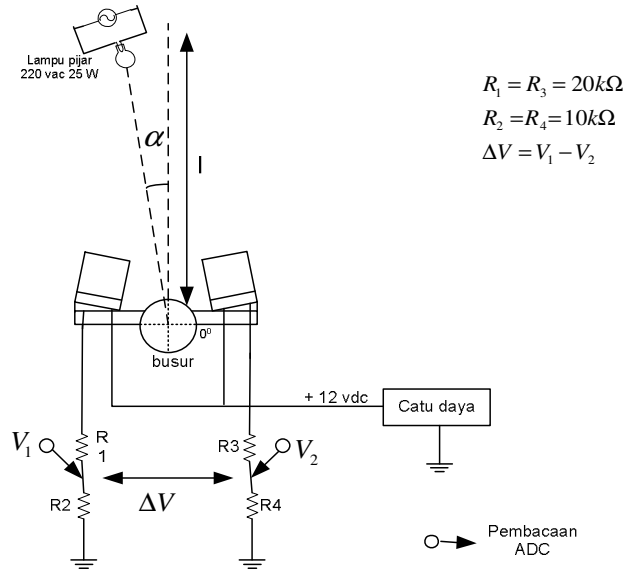
- modul sensor dan busur derajat
- modul mikrokontroler dengan program pembacaan ADC
- catudaya 5 vdc
- multimeter
- lampu pijar 25 watt

##### 3. Metode pengujian

Pengujian ini dilakukan untuk mengetahui hubungan antara perubahan tegangan pada sensor dengan perubahan posisi sudut sensor. Pada pengujian pertama dihitung banyaknya putaran motor setiap kenaikan  $1^{\circ}$ . Pada pengujian kedua, sensor akan dikenai cahaya dari lampu pijar dengan intensitas tertentu

dan posisi sudut sensor dibaca dengan menggunakan busur yang dipasang pada lengan sensor dan mengukur tegangan pada kedua sensor.

Gambar rangkaian pengujian :



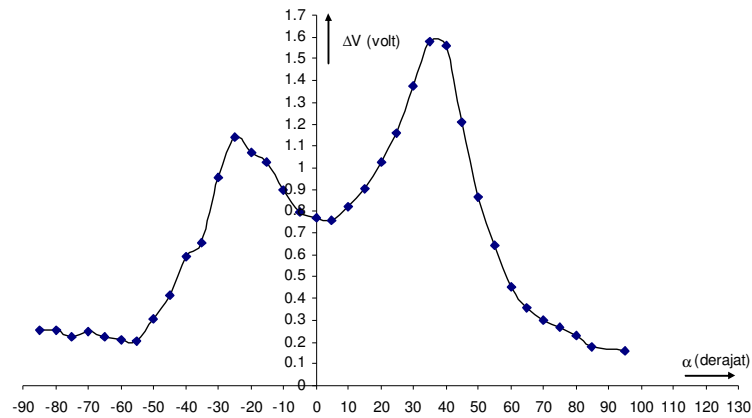
Gambar 4.1 Metode pengujian karakteristik sensor

#### 4. Prosedur pengujian

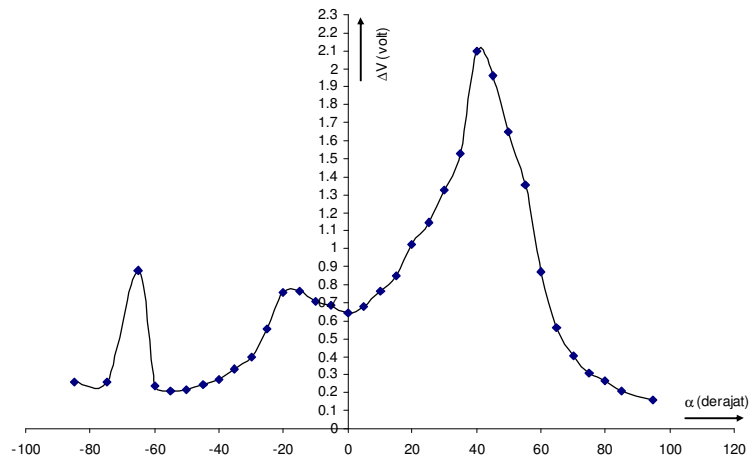
- a. Menggerakkan lengan sensor sejauh  $1^\circ$  dan hitung nilai jumlah putaran sensor.
- b. 1. Pengujian dilakukan di tempat gelap dengan sumber cahaya hanya lampu pijar.
2. Lampu pijar 25 watt sebagai sumber cahaya dengan jarak ( $l$ ) 22, 30, 60, 62.5 dan 90 cm.
3. Posisi  $0^\circ$  merupakan posisi lengan sensor tegak lurus dengan sumber cahaya dan memutar posisi sensor setiap  $10^\circ$
5. Mencatat hasil pembacaan ADC dan pembacaan multimeter

#### 5. Hasil pengujian

Rasio gir yang digunakan adalah 1:60. Dari data hasil pengujian pada lampiran 1, dapat diplot grafik 4.2 dan 4.3



Gambar 4.2. Grafik perubahan posisi sensor pada  $l = 62.5$  cm



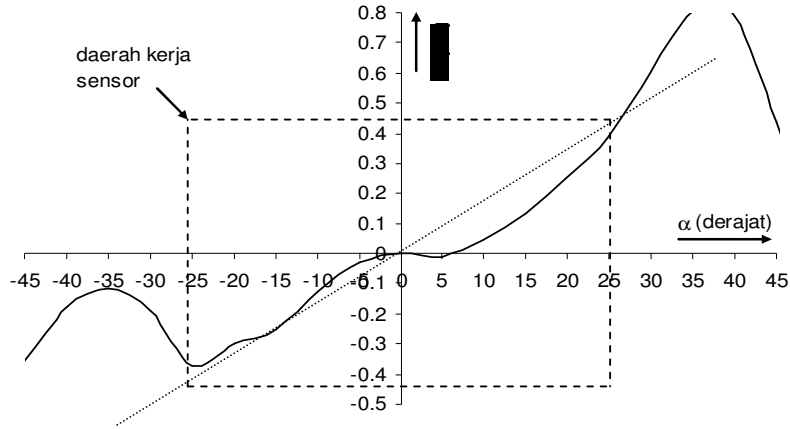
Gambar 4.3 Grafik perubahan posisi sensor pada  $l = 22$  cm

## 6. Analisa

Pengujian ini didapatkan rasio perbandingan gir yang digunakan pada motor mempunyai perbandingan setiap pergerakan  $6^0$  lengan sensor adalah jumlah 1 putaran motor. Sehingga didapatkan kesimpulan rasio gir adalah 60:1.

Dari pengujian yang dilakukan didapatkan nilai perbedaan nilai tegangan sebesar lebih kurang 0.771 volt pada kedua sensor pada posisi tegak lurus sumber cahaya. Pada penggunaan sensor pada sistem dilakukan kalibrasi

dengan menambahkan nilai tegangan yang terbaca pada sensor kiri dengan konstanta 0.771 volt.

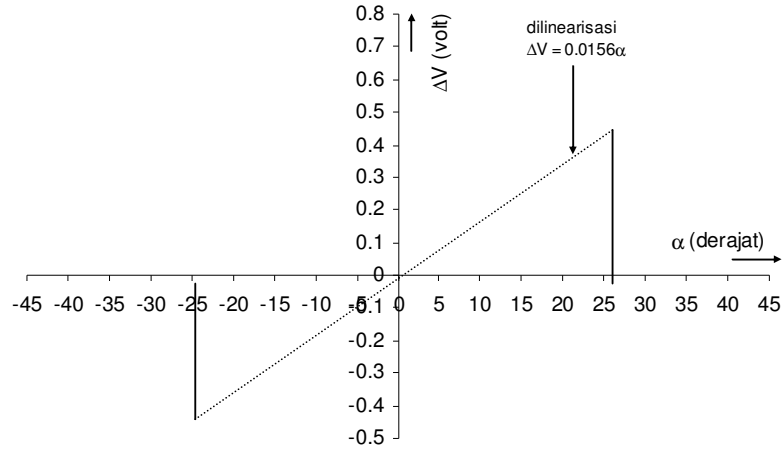


Gambar 4.4. Grafik daerah kerja sistem

Dari pembacaan grafik 4.4 , maka daerah yang dapat dilinearisasikan dengan dengan persamaan  $\Delta V = 0.0156\alpha$  pada daerah  $-25^\circ < \alpha < 25^\circ$  dan  $-0.353 < \Delta V < 0.399$  seperti terlihat pada gambar diatas. Sehingga, sistem ini mempunyai daerah kerja  $\pm 25^\circ$  dari sumber cahaya. Sehingga berdasarkan karakteristik sensor tersebut, pada rancangan ini digunakan persamaan masukan dari sistem sebagai berikut :

$$K_\alpha(s) = \begin{cases} 0 & \text{pada } \alpha < -25^\circ \text{ atau } \alpha > 25^\circ \\ 0.0156\alpha_{(s)} & \text{pada } -25^\circ < \alpha < 25^\circ \end{cases}$$

Dengan daerah kerja, seperti terlihat pada gambar 4.5. dibawah ini



Gambar 4.5. Grafik persamaan masukan dari sistem

#### 4.6. Pengujian dan analisa modul motor

##### 1. Tujuan

mengetahui karakteristik motor dan mendapatkan nilai konstanta  $k_m$  dan  $T_m$  motor

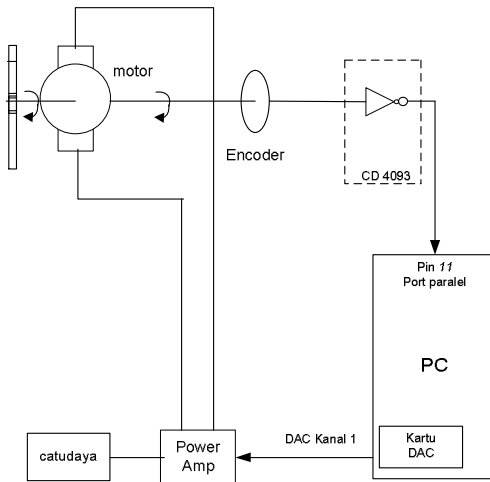
##### 2. Alat dan bahan yang digunakan

- modul motor dan penguat arus
- catu daya  $\pm 12$  VDC
- komputer yang terpasang kartu DAC *Contec* 12 bit dengan program turbo C++
- kabel paralel
- rangkaian sensor kecepatan

##### 3. Metode pengujian

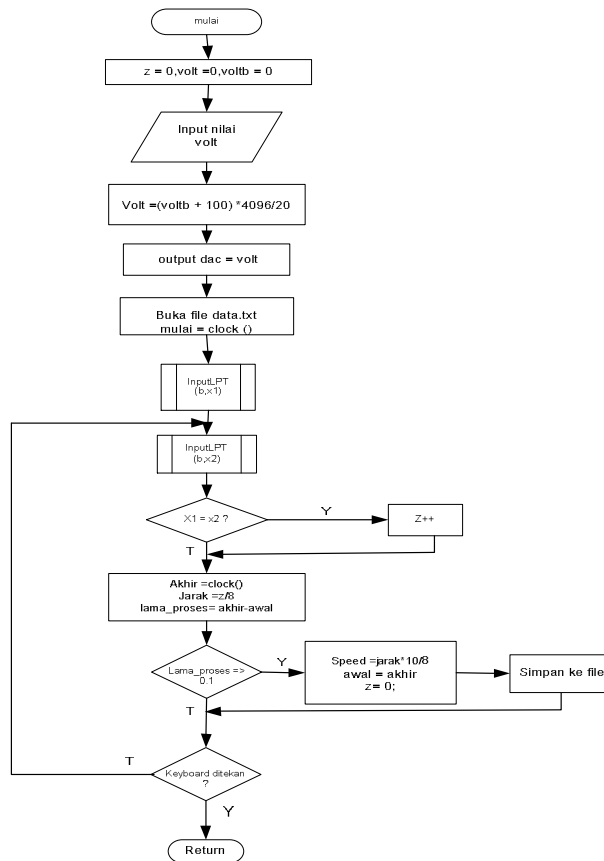
Pengujian dilakukan dengan memberi tegangan DC seketika pada motor dan merekam kecepatan motor setiap saat pada komputer yang telah terhubung dengan sensor piringan bercelah yang terpasang pada motor.

Rangkaian pengujian :



Gambar 4.6. Hubungan rangkaian pengujian respon motor

Kode bahasa C++ untuk pengujian ini dapat dilihat pada lampiran 6 . Diagram alir program tersebut dapat dilihat pada gambar 4.7.



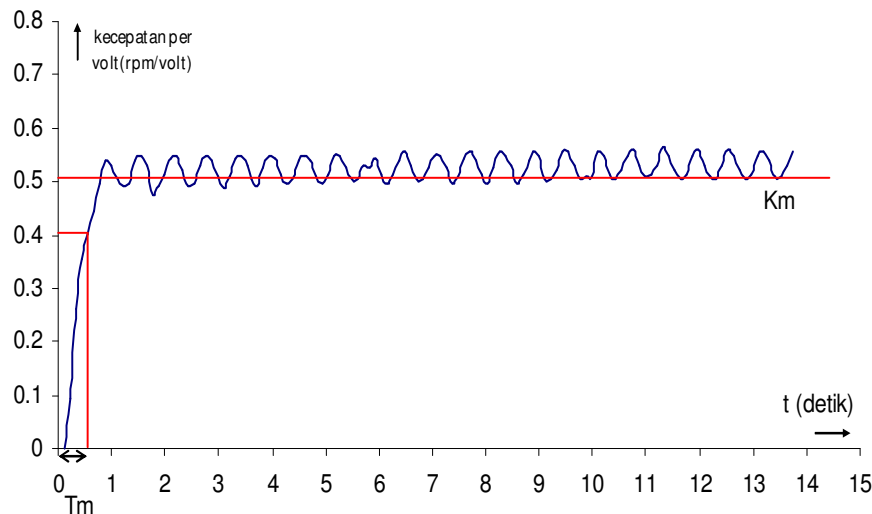
Gambar 4.7. Diagram alir pembacaan kecepatan motor dengan PC

#### 4. Prosedur pengujian

- a. Menghubungkan kanal 0 DAC contec dengan modul pengujian
- b. Mengeksekusi program .
- c. Memberi masukan nilai tegangan 2.0 VDC ke DAC
- d. Merekam hasil pembacaan selama lebih kurang 30 detik.
- e. Menyimpan hasil pembacaan di *d:\data.dat* Ulangi percobaan diatas untuk setiap kenaikan tegangan 0.1 VDC

#### 5. Hasil pengujian

Dengan data hasil pengujian dapat dilihat pada lampiran 2, dapat diplot grafik karakteristik kecepatan terhadap waktu pada saat tegangan input motor adalah 3,81 volt, seperti gambar 4.8.



Gambar 4.8 . Grafik  $K_m$  dan  $T_m$  motor

#### 7. Analisa

Konstanta penguatan ( $K_m$ ) dan konstanta waktu motor ( $T_m$ ) diperoleh secara grafis dari percobaan dengan sistem lup terbuka. Dari pembacaan grafik (lampiran 6) dan dengan

menggunakan  $\theta^o = E_a \times K_m \times \left[ 1 - e^{-\frac{t}{T_m}} \right]$  (persamaan 2-50), maka didapat nilai  $K_m =$

0,532. Dan  $T_m$  adalah waktu yang dibutuhkan saat  $\theta^o = 0.63 \times E_a \times K_m$  (persamaan 2-51),

maka didapat nilai  $T_m$  dari pembacaan grafik sebesar  $T_m = 0,41$  . Nilai  $K_m$  dan  $T_m$  digunakan dalam perhitungan pengaturan konstanta penguatan (K).

#### 4.3. Pengujian dan analisa ADC pada mikrokontroler

##### 1. Tujuan

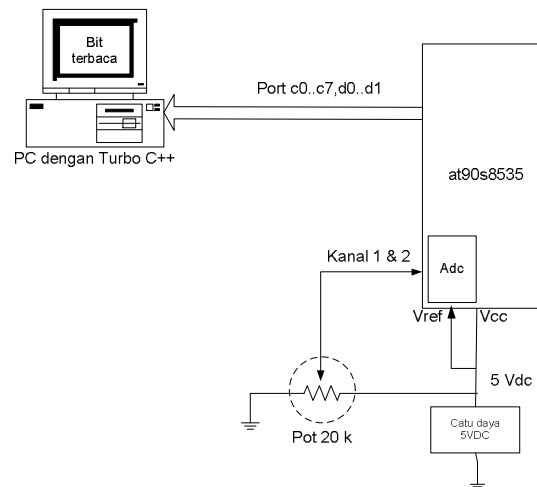
mengetahui cara kerja dan karakteristik linearitas ADC AT90S8535

##### 2. Alat dan bahan digunakan

- modul mikrontroller
- multimeter
- catu daya 5 VDC
- kabel paralel port
- komputer dengan program turbo C++

##### 3. Metode pengujian

Pengujian dilakukan dengan memberikan tegangan yang berubah pada modul adc dan membaca nilai bit-bit yang dikonversikan modul adc pada komputer dengan perantara paralel port. Pada mikrokontroler berisi program pembacaan input ADC dan program pembacaan hasil yang dibuat dengan bahasa C. Pengujian ini dilakukan dengan menggunakan kanal ADC yang berbeda dan tegangan referensi ( $V_{ref}$ ) yang sama.



Gambar 4.9. Blok diagram pengujian ADC

Kode assembly dan kode C program pengujian dapat dilihat pada lampiran 6

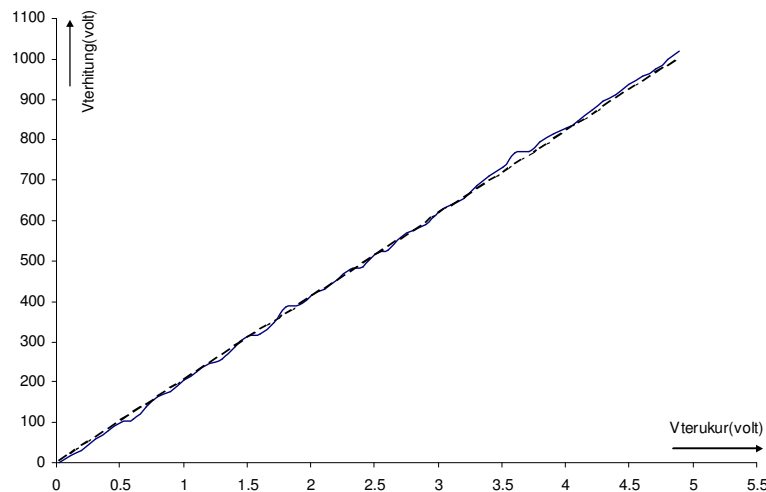


#### 4. Prosedur pengujian

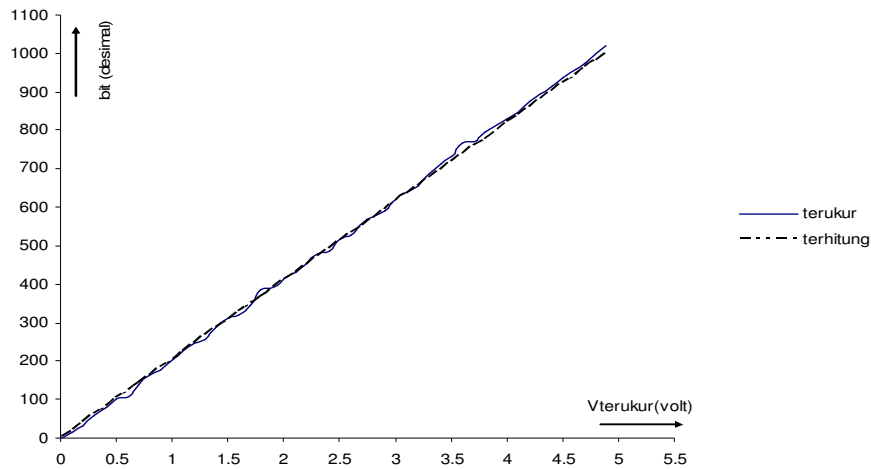
1. Menyimpan kode assembli untuk pembacaan pada kanal 0 adc
2. Menjalankan rangkaian modul adc pada mikrokontroler dan set tegangan referensi 5 vdc
3. Memvariasikasikan nilai tegangan dari 0 – 5 VDC setiap kenaikan 0.1 V
4. Mencatat nilai pembacaan pada PC
5. Mengulangi langkah 1 – 5 dengan pembacaan adc pada kanal 1
6. Menghitung % error dengan  $\%error = \left| \frac{pembacaan - perhitungan}{perhitungan} \right| \times 100\%$

#### 5. Hasil Pengujian

Dari data hasil pengujian yang terdapat pada lampiran 3, dapat diplot grafik linearitas ADC kanal 0 dan kanal 1 pada gambar 4.10 dan 4.11.



Gambar 4.10 Grafik linearitas ADC kanal 0



Gambar 4.11. Grafik linearitas ADC kanal 1

## 6. Analisa

ADC yang digunakan memiliki nilai kesalahan untuk kanal 1 sebesar 1,9372 % dan kanal 2 sebesar 1,801 %. Ini diakibatkan karena pembacaan alat ukur yang tidak tepat dan penggunaan resistor variabel sebagai pembagi tegangan yang memiliki nilai resistansi yang berubah-ubah. Dengan % kesalahan yang sangat kecil maka modul ini dapat bekerja dengan baik.

### 4.4. Pengujian dan analisa modul DAC

#### 1. Tujuan

mengetahui karakteristik linearitas modul DAC

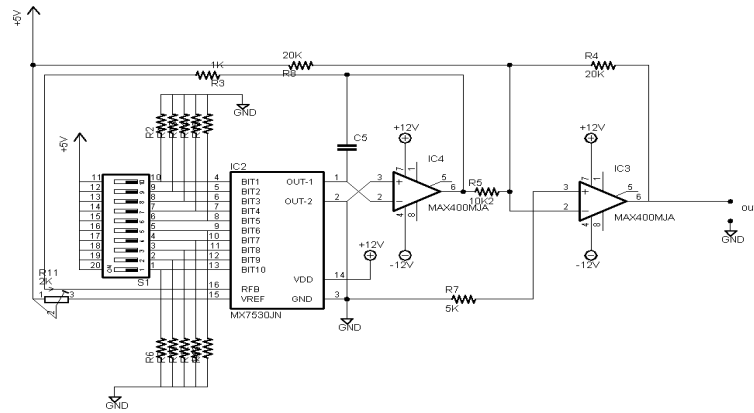
#### 2. Alat dan bahan yang digunakan

- Modul DAC
- Catu daya  $\pm 12$  VDC dan +5 VDC
- Multimeter
- Dip switch 10 kanal

#### 3. Metode pengujian

Pengujian dilakukan dengan memberikan bit-bit melalui dip switch dan mengukur tegangan keluaran modul DAC dengan 2 tegangan referensi yang berbeda.

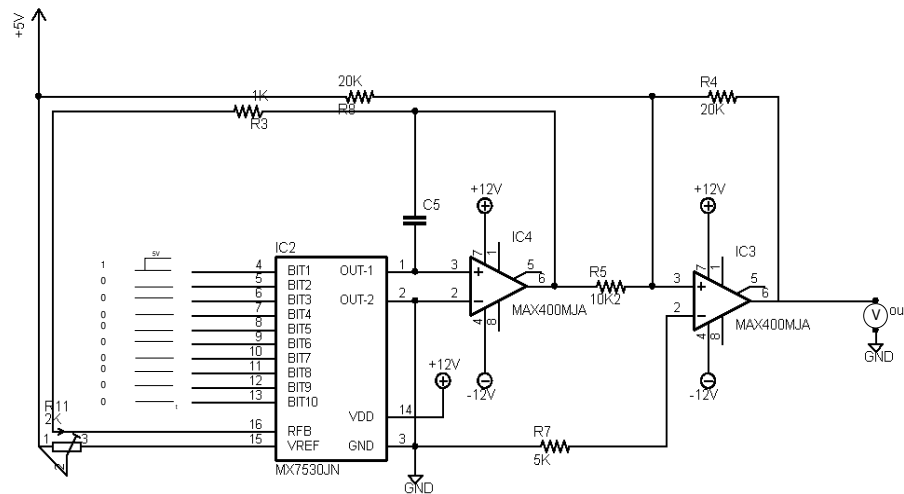
Rangkaian pengujian dapat dilihat pada gambar 4.11.



Gambar 4.11. Skema rangkaian pengujian

#### 4. Prosedur pengujian

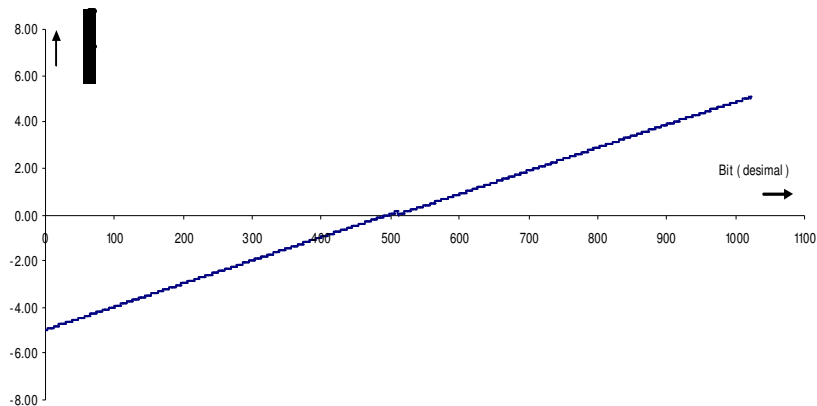
1. Menset tegangan referensi DAC +5 VDC
2. Mengkombinasikan saklar kombinasi setiap kenaikan 1 bit seperti terlihat pada gambar 4.12 dan mencatat data tegangan terukur



Gambar 4.12. Pengujian dengan input bit 0000000001

#### 5. Hasil pengujian

Data hasil pengujian dapat dilihat pada lampiran 4 dan dapat diplot grafik 4.13. yang memperlihatkan hasil pengujian dengan tegangan referensi +5 VDC .



Gambar 4.13 Grafik linearitas DAC pada tegangan referensi 5volt

## 6 . Analisa

Modul dac yang digunakan bekerja dengan baik untuk tegangan referensi +5 vdc.

### 4.5. Pengujian dan analisa modul komunikasi data serial

#### 1. Tujuan

Mengetahui prinsip kerja modul UART pada mikrokontroler .

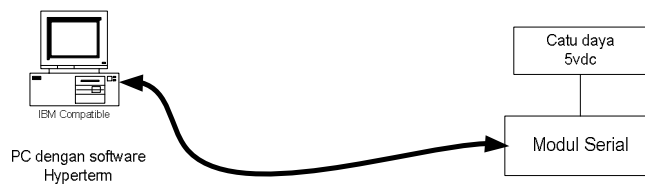
#### 2. Alat dan bahan yang digunakan

- PC dengan OS Windows dengan program “*hyperterm*”
- Modul serial dan mikrokontroler
- Catu daya 5 VDC
- Kabel serial dengan panjang 45 cm

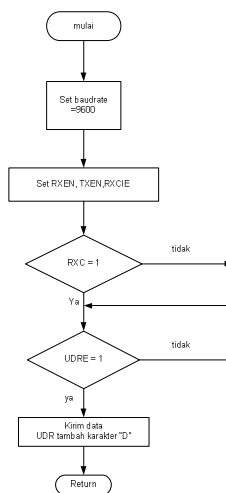
#### 3. Metode pengujian

Pengujian dilakukan dengan mengirimkan karakter ke modul serial dari PC ke mikrokontroler. Mikrokontroler diprogram untuk mengembalikan karakter yang diterimanya, disertai dengan karakter penanda hubungan. Pengujian ini dinamakan dengan pengujian loopback.

Gambar rangkaian pengujian



. Kode assembly yang digunakan pada pengujian ini dapat dilihat pada lampiran 6. Diagram alir program tersebut dapat dilihat pada gambar 4.14.



Gambar 4.14 Diagram alir program pengujian modul serial

#### 4. Prosedur pengujian

- a. Menyimpan program test\_serial dalam format hex ke mikrokontroler
- b. Mengaktifkan rangkaian modul serial dan mikrokontroler
- c. Menjalankan program terminal “Hyperterm” dengan setting baudrate 9600, 8 bit data, tanpa bit paritas dan 1 stop bit.
- d. Mengetikkan kata “FAJRIL AKBAR” per huruf pada hyperterm dan mencatat hasil pembacaan pada program hyperterm

#### 4. Hasil pengujian

Hasil pengujian dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil pengujian modul serial

Dikirim PC	Diterima PC
F	FD
A	AD
J	JD

R	RD
I	ID
L	LD
	D
A	AD
K	KD
B	BD
A	AD
R	RD

#### 6. Analisa

Setiap karakter yang dikirimkan dikirimkan kembali dengan benar oleh modul serial. Dengan jarak transmisi serial yang cukup pendek dan mikrokontroler tidak melakukan pengolahan data yang banyak. Sehingga modul serial dapat bekerja dengan benar .

#### 4.7. Pengujian sistem

##### 1. Tujuan

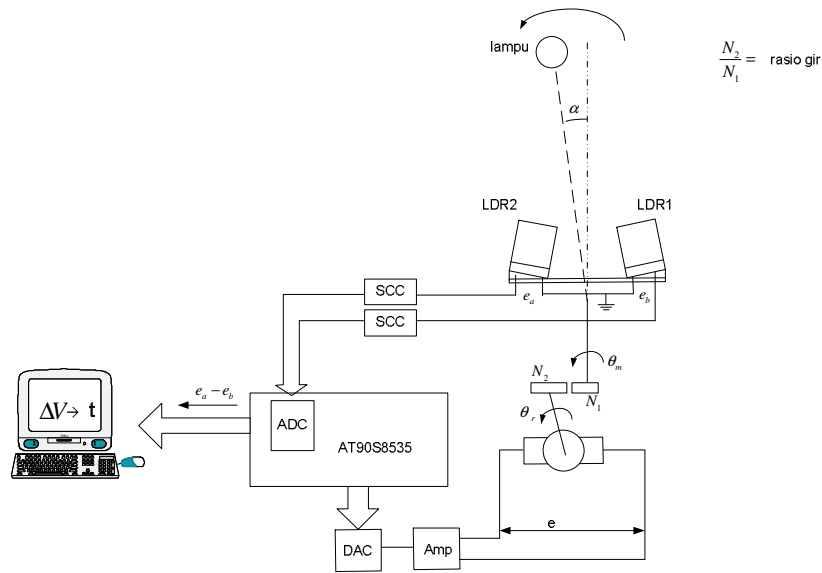
mengetahui pengaruh perubahan konstanta k terhadap performansi sistem .

##### 2. Alat dan bahan yang digunakan

- Catu daya  $\pm 12$  dan 5 VDC
- Penggerak kolektor energi surya
- Komputer dengan program turbo C++
- Kabel serial

##### 3. Metode pengujian

pengujian dilakukan dengan mengubah nilai konstanta k, dan mencatat data tegangan motor terhadap waktu



Gambar 4.15 Blok diagram pengujian system

#### 4. Hasil pengujian

Data-data hasil pengujian dapat dilihat pada lampiran 5.

#### 5. Analisa

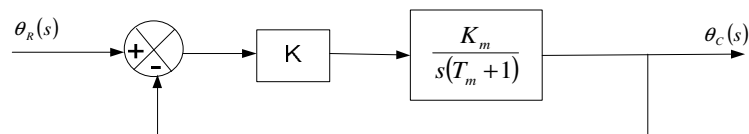
Untuk mengontrol sebuah motor maka harus diketahui karakteristik dari motor tersebut. Dengan menentukan nilai perbandingan antara kecepatan dengan tegangan yang diberikan pada motor dinyatakan sebagai  $K_m$ , sedangkan konstanta waktu motor dinyatakan sebagai  $T_m$ . Dengan melalui percobaan-percobaan yang terdapat pada lampiran maka didapat :

$$K_m = 0.532$$

$$T_m = 0.41$$

### 6.1. Respon pengaturan posisi motor DC

Posisi motor dapat diatur dengan menggunakan sistem pengaturan lup tertutup.

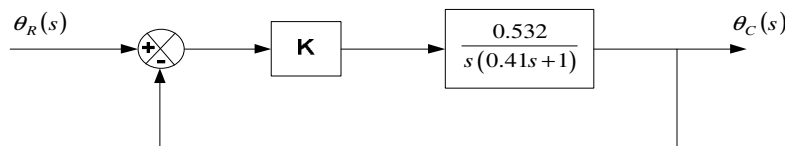


Gambar 4.16 Sistem motor DC dengan konstanta K

Untuk memperbaiki respon sistem, maka digunakanlah pengaturan konstanta penguatan ( $K$ ). Berdasarkan persamaan (2-29), maka diperoleh :

$$\frac{\theta_m(s)}{E_a(s)} = \frac{0,532K}{0,41s^2 + s + 0,532K}$$

dan dengan memasukkan nilai  $K_m$  dan  $T_m$  maka blok diagram sistem tertutup rancangan adalah pada gambar berikut :



Gambar 4.17. Rancangan dengan konstanta  $K$

Agar dihasilkan respon sistem yang paling baik pada  $\zeta = 0,8$  dapat dilakukan dengan mengatur penguatan,  $K$ . Secara grafis pengaturan  $K$  artinya mengatur penempatan pole pada bidang kompleks sehingga diperoleh sistem yang memenuhi performansi yang diinginkan. Dengan mempergunakan metode kedudukan akar, maka posisi seluruh kutub lup tertutup dapat diketahui.

Dari fungsi alih lup tertutup sistem, maka :

$$s_{12} = -\frac{1}{2(0,41)} \pm \sqrt{\frac{1 - 4(0,41)(0,532)K}{4(0,41)^2}}$$

Sehingga diketahui bahwa sistem memiliki dua buah kutub, yaitu di titik  $s_1 = -1,2195 + \sqrt{1,4872 - 1,2975K}$  dan titik  $s_2 = -1,2195 - \sqrt{1,4872 - 1,2975K}$  pada bidang  $s$ . Dengan mengatur nilai  $K$ , maka diperoleh tempat kedudukan akar sistem lup tertutup.

Dari tabel (4.2) dapat dilihat bahwa sistem dengan pengaturan konstanta penguat ( $K$ ) akan berada pada tiga daerah operasi, yaitu : redaman lebih, redaman kritis dan redaman kurang.

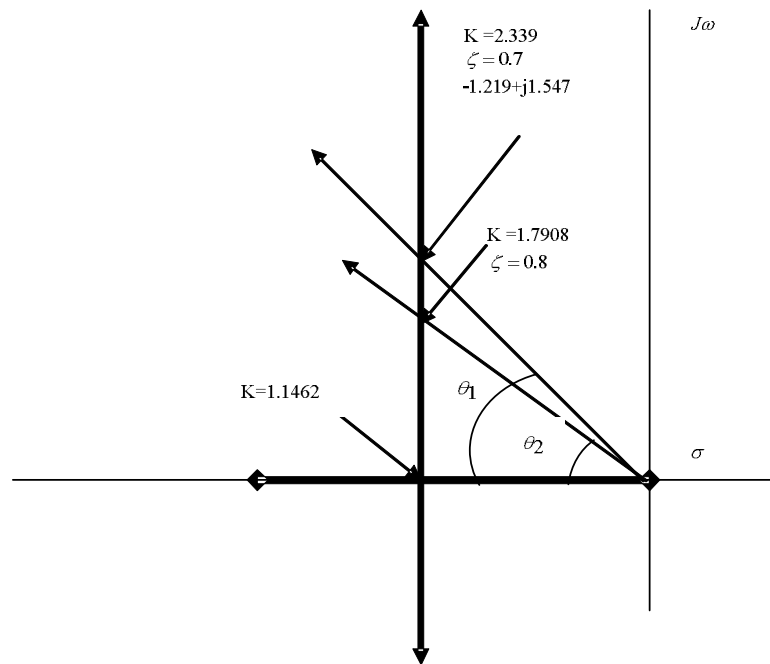
Daerah operasi redaman lebih terjadi pada  $0 < K < 1,1462$  hal ini akan mengakibatkan kutub-kutub akan bergeser sepanjang sumbu nyata. Daerah operasi redaman kritis terjadi bila konstanta penguatan  $K = 1,1462$ . Pada daerah redaman kritis ini kedua pole akan berhimpit pada satu titik di sumbu nyata. Sedangkan pada saat



konstanta penguatan  $K > 1,1462$  maka pole-pole akan bergerak di bidang imajiner, sehingga akan terjadi osilasi pada sistem.

Tabel 4.2. Respon sistem akibat perubahan nilai  $K$

K	Pole 1	Pole 2
0.05	-0.0269	-2.4121
0.1	-0.0583	-2.3807
0.5	-0.3039	-2.1351
0.7	-0.4586	-1.9803
0.9	-0.6544	-1.7846
1.0	-0.7844	-1.6550
1.1	-0.9747	-1.4643
1.2	-1.2195+j0.264	-1.219-j0.264
1.25	-1.2195+j0.366	-1.2195-j0.366
1.27	-1.2195+j0.400	-1.2195-j0.400
1.28	-1.2195+j0.416	-1.2195-j0.416
1.3	-1.2195+j0.461	-1.2195-j0.461
2.0	-1.2195+j1.052	-1.2195-j1.052
2.5	-1.2195+j1.325	-1.2195-j1.325
3.0	-1.2195+j1.550	-1.2195-j1.550
3.5	-1.2195+j1.747	-1.2195-j1.474
5.0	-1.2195+j2.236	-1.2195-j2.236



Gambar 4.18. Tempat kedudukan akar motor DC

Secara teoritis, konstanta penguatan  $K$  dapat dihitung dengan data percobaan ( $k_m = 0,532 \text{ rpm/volt}$ , dan  $T_m = 0,41 \text{ det}$ ). Dengan asumsi sistem yang terbaik akan memiliki konstanta redaman  $\zeta = 0,8$ . Dari persamaan  $\frac{\theta_m(s)}{E_a(s)} = \frac{0,532K}{0,41s^2 + s + 0,532K}$  yang identik dengan persamaan dibawah

$$\frac{\theta(s)}{E_a(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

dan didapat

$$\omega_n = \frac{1}{T_m 2\zeta}$$

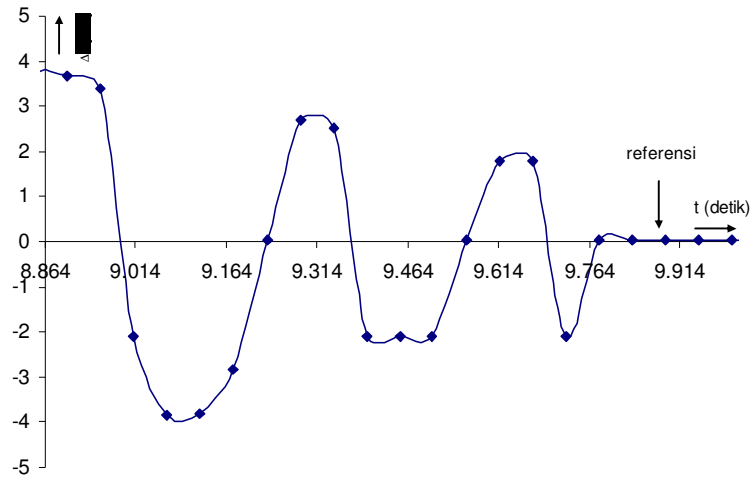
$$\omega_n = \frac{1}{0,41 \times 2 \times 0,8} = 1,524 (\text{rad/det}^2)$$

$$K = \frac{(1,524)^2 \times 0,41}{0,532} = 1,790$$

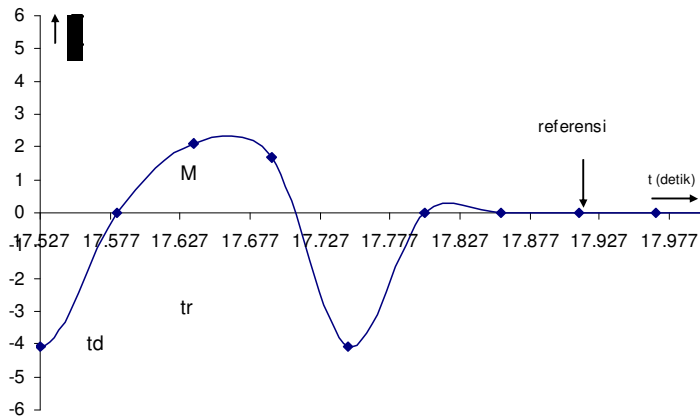
Pada pengujian, ini tidak dilakukan pengujian untuk nilai  $K$  yang bernilai pecahan. Ini tidak dapat penulis lakukan karena keterbatasan pengetahuan penulis tentang pemrograman assembly, dan dibutuhkan perencanaan dan perancangan program

yang lebih rumit dan kompleks. Untuk pengujian dilakukan dengan memvariasikan  $K$  yang bukan merupakan pecahan.

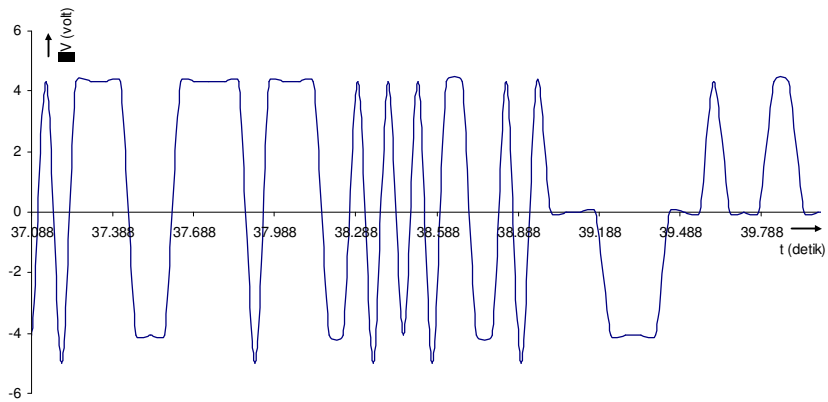
Dari grafik pada lampiran 5, dapat diplot grafik respon perubahan posisi motor dengan menggeser arah lampu, dengan nilai  $K$  yang berbeda.



Gambar 4.19. Respon sistem saat  $K = 1$

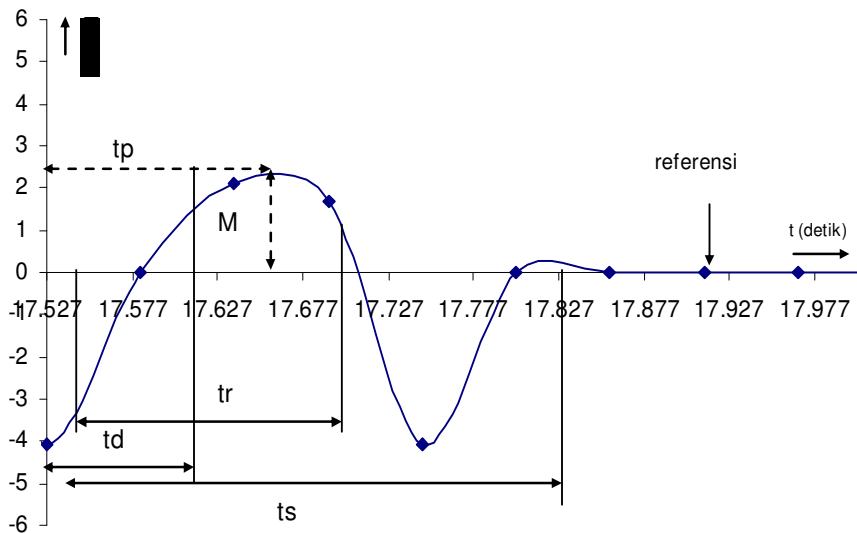


Gambar 4.20. Respon sistem saat  $K = 2$



Gambar 4.21. Respon sistem saat  $K = 4$

Pada pengujian karakteristik respon transien sistem, dilakukan pengujian dengan memvariasikan nilai konstanta  $K$ . Pada gambar dapat dilihat dengan memperbesar nilai  $K$ , sistem akan tidak stabil. Dengan mempergunakan nilai  $K = 2$ , dengan  $\zeta = 0.757$ , maka didapatkan karakteristik respon transien sistem seperti terlihat pada gambar 4.20.



Gambar 4.22. Karakteristik respon saat  $K = 2$

- Lewatan Maksimum :

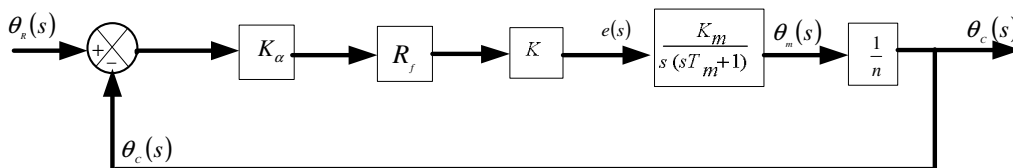
$$M = \%OS = \frac{2.5 - 0}{0} \times 100 = 25\%$$

- Waktu Puncak

- $t_p = 0.123$  detik
- Waktu naik
- $t_r = 0.153$  detik
- Waktu tunda
- $t_d = 0.3$  detik

## 6.2. Respon sistem penggerak kolektor surya

Untuk menentukan fungsi alih sistem secara menyeluruh sehingga didapatkan hubungan antara sudut terjadi dengan sudut yang diinginkan.



Gambar 4.23. Blok diagram sistem tanpa gangguan

Keterangan :

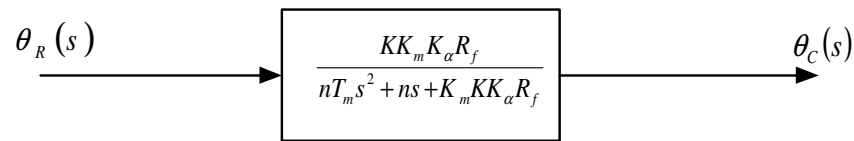
- $\theta_R(s)$  = Sudut setting antara sensor terhadap sumber cahaya
- $\theta_c(s)$  = Sudut yang terjadi antara sensor terhadap sumber cahaya
- $K_\alpha$  = konstanta hubungan beda tegangan sensor dan posisi sudut
- $R_f$  = rasio penguatan tegangan sensor
- $K_m$  = konstanta penguatan motor
- $T_m$  = konstanta penguatan motor
- $n$  = rasio gir motor penggerak

Fungsi alih dari blok diagram lup tertutup diatas adalah sebagai berikut :

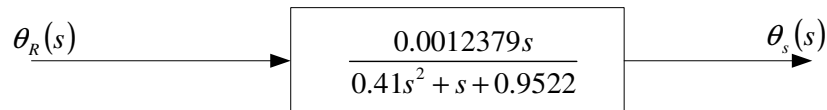
$$\frac{\theta_R(s)}{\theta_c(s)} = \frac{KK_m K_\alpha R_f}{nT_m s^2 + ns + K_m K K_\alpha R_f}$$

Berdasarkan hasil pengujian sebelumnya didapatkan nilai konstanta input dari sistem adalah  $x K_\alpha = 0.0156\theta$  pada  $-25^0 < \theta < 25^0$ . Sehingga dapat dibuat blok diagram

lup terbuka fungsi alih untuk keadaan diatas yang menyatakan perbandingan  $\theta_i$  dan  $\theta_c$  yaitu :



Untuk menganalisa dan menentukan tempat kedudukan akar pada persamaan sistem secara menyeluruh maka sistem blok diagram yang dipakai adalah blok diagram sistem tanpa gangguan dan dibentuklah sebuah lup secara keseluruhan dengan memasukkan nilai konstanta  $R_f = 5$ ,  $K_m = 0.532$ ,  $T_m = 0.41$ ,  $K = 1,790$  saat  $\zeta=0.8$ , maka blok diagram sistem adalah :



Gambar 4.24. Blok diagram sistem setelah substitusi konstanta

Sehingga didapatkan persamaan sistem adalah :

$$\frac{\theta_s}{\theta_R} = \frac{0.001279s}{0.41s^2 + 1.0012379s + 0.9522}$$

## Bab 5

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan analisa hasil penelitian dapat disimpulkan bahwa:

1. Rancangan sistem kontrol posisi penggerak kolektor energi surya dengan menggunakan mikrokontroler dengan penerapan kontroler proporsional mempunyai karakteristik respon transien yang cukup baik.

2. Pada rancangan sistem ini, karena keterbatasan pengetahuan penulis dalam penguasaan bahasa assembly, belum dapat memvariasikan nilai K untuk bilangan berkoma.

## 5.2 Saran

1. Untuk memperbaiki respon sistem sebaiknya digunakan kontroler PI, PD, PID, lag, lead atau lag-lead.
2. Untuk penelitian selanjutnya agar dapat memvariasikan nilai K dalam rentang nilai yang lebih kecil.
3. Penggunaan bahasa pemrograman lain seperti C sebagai pengganti bahasa assembly akan lebih mempermudah dalam pembuatan program pada mikrokontroler.

## DAFTAR PUSTAKA

1. Nur, Erinaldi, *Perancangan Suatu Sistem Kontrol Posisi untuk Penggerak Kolektor Energi Surya*, Universitas Andalas, Padang, 1999
2. Ogata, Katshuhiko, *Teknik Kontrol Automatik (Sistem Pengaturan) jilid 1*, Erlangga, Jakarta, 1995
3. Kuo, Benjamin C., *Automatic Control Systems-7<sup>th</sup> Edition*, Prentice Hall, New Jersey, 1995.
4. Atmel Corp, *Datasheet Microcontroller AT90S8535-8P*, Atmel Corp, 2001, <http://www.atmel.com/products/>
5. Atmel Corp, *Appliaction Notes AVR*, Atmel Corp, 2001, <http://www.avrfreaks.com/>

## Lampiran Program

```
;/*****/
; Project      : Program penggerak kolektor surya dengan AT90S8535
; Version      : 1.0
; Date        : 1/13/2004
; Author       : Fajril Akbar
; Chip type    : AT90S8535
; Clock frequency : 8.000000 MHz
; Memory model : Small
; External SRAM size : 0
; Comments     :
;
;/*****/

.include "8535def.inc"
.listmac

    .macro comp
        cpi  r26,low(@0)
        ldi  r30,high(@0)
        cpc  r27,r30
    .endm

    .macro outw
        clr  r26
        clr  r27
        ldi  r26,low(@0)
        ldi  r27,high(@0)
    .endm

;-----
;
; peta memori SRAM yang dialokasikan untuk program
;
;-----

.def  remote      = r25
.equ  adc_2ch      = 0x60 ; 12 bytes
.equ  konst_k      = 0x6D ; 4 bytes
.equ  adc_data     = 0x71 ; 4 bytes
.equ  buffer       = 0x79 ; 8 bytes  buffer serial

;-----
; variabel baudrate UART
;-----
.equ  baudrate     = 0x33

;-----
; variabel kalibrasi dac
;-----
.equ  dac_nol      = 0x200

;-----
; Variabel tegangan referensi DAC
;-----
.equ  vref_dac     = 0x3FF

;-----
; Peta memory EEPROM untuk konfigurasi
```



```

;-----
.equ   konf_eeprom      = 0                               ;8 bytes

;-----
;
;
; Vektor interupsi
;
;-----
.cseg
.org 0x0000
    rjmp  inisialisasi
    reti                               ; INT0
    reti                               ; INT1
    reti                               ; TIMER/COUNTER2 Compare Match
    reti                               ; TIMER/COUNTER2 Overflow
    reti                               ; TIMER/COUNTER1 Capture Event
    reti                               ; TIMER/COUNTER1 Capture Match A
    reti                               ; TIMER/COUNTER1 Capture Match B
    reti                               ; TIMER/COUNTER1 Overflow
    reti                               ; TIMER/COUNTER0 Overflow
    reti                               ; SPI, Serial Transfer Complete
    reti                               ; UART, RX Complete
    reti                               ; UART, Data Register Empty
    reti                               ; UART, TX Complete
    rjmp  int_adc                ; ADC Conversion Complete
    reti                               ; EEPROM Ready
    reti                               ; Analog Comparator

;-----
;
;
; Interupsi dari adc
;
;-----
int_adc:
    in    r16,mcucr
    andi  r16,0xff^(1<<se)
    out   mcucr,r16
    in    r17,adcl
    in    r18,adch
    reti

;-----
;
;
; Fungsi inisialisasi hardware
;
;-----
inisialisasi:
    cli
    clr   r30
    out   EECR,r30
    out   MCUCR,r30
    ldi   r16,low(RAMEND)
    out   SPL,r16
    ldi   r16,high(RAMEND)
    out   SPH,r16
    ldi   r16,0x80
    out   ACSR,r16
    ldi   r16,0xFF
    out   DDRB,r16
    ldi   r16,0x00
    out   PORTB,r16
    ldi   r16,0xFF

```

```
    out    DDRC,r16
    ldi    r16,0x00
    out    PORTC,r16
    ldi    r16,0xF0
    out    DDRD,r16
    ldi    r16,baudrate
    out    UBRR,r16

;-----
;
; Program utama
;
;-----

main :
    rcall  baca_konf

    clr    remote

loop_main:
    rcall  cek_inst
    sbrs  remote,1
    rcall  proses_baca
    rjmp  loop_main

;-----
;
; Fungsi baca eeprom
;
;-----
EEPRead:
    sbic  EECR,EWE
    rjmp  EEPRead
    out   EEARL,r17
    sbi   EECR,EERE
    in    r18,EEDR
    ret

;-----
;
; Fungsi tulis eeprom
;
;-----
EEPWrite:
    sbic  EECR,EWE
    rjmp  EEPWrite
    out   EEARL,r16
    out   EEDR,r17
    sbi   EECR,EEMWE
    sbi   EECR,EWE
    ret

;-----
;
; fungsi loading konfigurasi dari EEPROM untuk menyimpan nilai k
;
;-----
baca_konf:
    ldi    r19,4
    ldi    r17,low(konf_eeprom)
    ldi    ZL,low(konst_k)
    rcall  EEPRead
    st     Z+,r18
```

```
konf:
    inc    r17
    rcall  EEPRead
    st     Z+,r18
    dec    r19
    cpi    r19,1
    brne  konf
    ret

;-----
;
; Fungsi kirim karakter uart
;
;-----
sendchr:
    sbi    UCR, TXEN
    sbis   USR, UDRE
    rjmp   sendchr
    out    UDR,r17
    cbi    UCR, TXEN
    ret

;-----
;
; fungsi rekam karakter uart
;
;-----
recchr:
    sbi    UCR, RXEN
    sbis   USR, RXC
    rjmp   recchr
    in     r17,UDR
    cbi    UCR, RXEN
    ret

;-----
;
; Program pembacaan ADC
;
;-----
proses_baca:
    clr    r16
    ldi    r19,0                ;kanal 0
    ldi    r20,0
    rcall  baca_ch
    in     r6,adcl
    in     r7,adch
    ldi    r19,1                ;kanal 1
    ldi    r20,2
    rcall  baca_ch
    in     r8,adcl
    in     r9,adch
    ldi    ZL,low(adc_data)
    st     Z+,r6
    st     Z+,r7
    st     Z+,r8
    st     Z+,r9
    rjmp   proses
    ret
baca_adc:
    sbic   adcsr, adsc
    rjmp   baca_adc
    ldi    r16,0x8F
    out    adcsr,r16
```

```
    in    r16,mcucr
    ori   r16,(1<<se)
    out   mcucr,r16
    sei
    sleep
    cbi   adcsr,adie
    clr   r16
    ret

baca_ch:
    out   admux,r19
    rcall baca_adc
    ldi   ZL,low(adc_data)
    add   ZL,r20
    st    Z+,r17
    st    Z+,r18
    ret

;-----
;
; Fungsi pemrosesan output ADC
;
;-----
proses:
    mov   XL,r6
    mov   XH,r7
    mov   YL,r8
    mov   YH,r9
    subi  XL,low(-158)
    sbci  XH,high(-158)
    cp    XL,YL
    cpc   XH,YH
    brge  putar_kanan

;-----
;
; Fungsi kontrol manual
;
;-----
putar_kiri:
    sub   YL,XL
    sbc   YH,XH
    cpi   YL,low(0x52)
    ldi   r16,high(0x52)
    cpc   YH,r16
    brge  out_try
    lsl   YL
    rol   YH
    outw  dac_nol
    sub   XL,YL
    sbc   XH,YH
    rjmp  proses_1

putar_kanan:
    sub   XL,YL
    sbc   XH,YH
    comp  0x0052
    brge  out_try
    lsl   XL
    rol   XH
    subi  XL,low(-dac_nol)
    sbci  XH,high(-dac_nol)
    rjmp  proses_1
```

```
;-----  
;  
; Fungsi pemrosesan ADC  
; Vout = konst_k x (Vkanal1 - Vkanal2)  
;  
;-----  
  
proses_1:  
    comp    dac_nol  
    brge    great  
    comp    0x62  
    brlt    dac_4  
    comp    0x13E  
    brge    dac_2  
    rjmp    out_dac  
  
great:  
    comp    0x2BE  
    brlt    dac_1  
    comp    0x391  
    brge    dac_3  
    rjmp    out_dac  
  
dac_1:  
    comp    0x246  
    brlt    out_nol  
    outw    0x2BE  
    rjmp    out_dac  
  
dac_2:  
    comp    0x1AC  
    brge    out_nol  
    outw    0x13E  
    rjmp    out_dac  
  
dac_3:  
    outw    0x391  
    rjmp    out_dac  
  
dac_4:  
    outw    0x62  
    rjmp    out_dac  
  
out_try:  
    outw    dac_nol  
    rjmp    out_dac  
  
out_dac:  
    lsl    r27  
    lsl    r27  
    lsl    r27  
    lsl    r27  
    lsl    r27  
    lsl    r27  
    out    PORTC,r26  
    out    PORTB,r27  
    cbr    r20,128  
    out    PORTD,r20  
    ret  
  
out_nol:  
    clr    r20  
    sbr    r20,128  
    out    PORTD,r20
```

```
        outw   dac_nol
        rjmp   out_dac
        ret

manual_kiri:
        outw   0x12A
        rcall  out_dac
        rcall  loop
        rcall  proses_baca
        rcall  simpan_data
        rcall  kirim2pc
        rcall  out_nol
        ret

manual_kanan:
        outw   0x2C8
        rcall  out_dac
        rcall  loop
        rcall  proses_baca
        rcall  simpan_data
        rcall  kirim2pc
        rcall  out_nol
        ret

;-----
;
; Fungsi cek instruksi dari PC
;
;-----
cek_inst:
        sbi    UCR,RXEN
        sbis   USR,RXC
        rjmp   not_remote
        in     r17,UDR
        cbi    USR,RXEN

parsing:
        cpi    r17,'S'
        breq   mulai_remote
        cpi    r17,'q'
        breq   akhiri_remote
        cpi    r17,'R'
        breq   kirim_konf
        cpi    r17,'W'
        breq   tulis_konf
        cpi    r17,'T'
        breq   start_pc
        cpi    r17,'r'
        breq   manual_kanan
        cpi    r17,'l'
        breq   manual_kiri
        ret

;-----
;
; Fungsi aktivasi proses
;
;-----

start_mikro:
```

```
        ldi    remote,1
        ret

stop_mikro:
        rcall  out_nol
        sbrc  remote,0
        sbrc  remote,1
        dec   remote
        ret

start_pc:
        inc   remote

stop_pc :
        rcall  proses_baca
        rcall  simpan_data
        rcall  kirim2pc
        sbi    UCR,RXEN
        in     r17,UDR
        cbi    USR,RXEN
        cpi    r17,'t'
        brne  stop_pc
        rcall  stop_mikro
        rcall  out_nol
        ret

not_remote:
        sbrs  remote,0
        sbrs  remote,1
        rcall  start_mikro
        ret

mulai_remote:
        clr   r20
        sbr   r20,64
        out   PORTD,r20
        rcall  stop_mikro
        ldi   remote,2
        ret

akhiri_remote:
        ldi   remote,1
        ret

;-----
;
; Fungsi baca konfigurasi
;
;-----

tulis_konf:
        ldi   XL,low(buffer)
        ldi   r18,0

tunggu:
        rcall  recchr
        st    X+,r17
        inc   r18
        cpi   r17,0xA
        brne  tunggu
        cpi   r18,0x9
        brne  error_konf
        ldi   XL,low(buffer)
        ldi   r22,low(konf_eeprom)
```

```
tulis_eep:
    ld    r17,Y+
    swap r17
    andi  r17,0xF0
    ld    r20,Y+
    andi  r20,0x0F
    or    r17,r20
    ld    r16,Y+
    swap r16
    andi  r16,0xF0
    ld    r20,Y+
    andi  r20,0x0F
    or    r16,r20
    clr   r18
    rcall BCD2bin16
    mov   r16,r22
    mov   r17,R14
    rcall EEPWrite
    inc   r22
    mov   r16,r22
    mov   r17,R15
    rcall EEPWrite
    inc   r22
    cpi   r22,low(konf_eeprom + 4)
    brne tulis_eep
    ret
```

```
error_konf:
    ldi   r17,'E'
    rcall sendchr
    ldi   r17,'R'
    rcall sendchr
    ret
```

```
;-----
;
; Fungsi kirim konfiugrasi
;
;-----
```

```
kirim_konf:
    ldi   r17,'P'
    rcall sendchr
    ldi   r17,':'
    rcall sendchr
    ldi   YL,low(konst_k)
    ldi   r20,0
```

```
baca_1 :
    inc   r20
    ld    r17,Y+
    ld    r16,Y+
    rcall bin2ascii
    mov   r17,R4
    rcall sendchr
    mov   r17,R3
    rcall sendchr
    mov   r17,R2
    rcall sendchr
    mov   r17,R1
    rcall sendchr
    ldi   r17,':'
    rcall sendchr
    cpi   r20,2
```



```
brne  baca_1
ldi   r17,0x0D
rcall sendchr
ldi   r17,0x0A
rcall sendchr
ret

;-----
;
; Fungsi menyimpan data adc dalam ascii
;
;-----
simpan_data:
    clr    r20
    ldi    YL,low(adc_data)
    ldi    XL,low(adc_2ch)
simpan_1:
    inc    r20
    ld     r16,Y+
    ld     r17,Y+
    rcall bin2ascii
    mov    r16,R5
    st     X+,r16
    mov    r16,R4
    st     X+,r16
    mov    r16,R3
    st     X+,r16
    mov    r16,R2
    st     X+,r16

    mov    r16,R1
    st     X+,r16
    ldi    r16,0x9
    st     X+,r16
    cpi    r20,2
    brne  simpan_1
    ret

;-----
;
; Fungsi kirim data ke PC
;
;-----
kirim2pc:
    ldi    r17,'D'
    rcall sendchr
    ldi    r18,12
    ldi    XL,low(adc_2ch)

loop:
    ld     r17,X+
    rcall sendchr
    dec    r18
    brne  loop
    ldi    r16,0x0D
    rcall sendchr
    ldi    r16,0x0A
    rcall sendchr
    ret

bin2ascii:
    ldi    r18,16
    clr    r15
    clr    r14
    clr    r13
```

```
        clr     ZH
bBCDx_1 :
        lsl     r16
        rol     r17
        rol     r13
        rol     r14
        rol     r15
        dec     r18
        brne   bBCDx_2
        mov     r16,r13
        andi   r16,0x0F
        ori    r16,0x30
        mov     r1,r16
        mov     r16,r13
        swap   r16
        andi   r16,0x0F
        ori    r16,0x30
        mov     r2,r16
        mov     r16,r14
        andi   r16,0x0F
        ori    r16,0x30
        mov     r3,r16
        mov     r16,r14
        swap   r16
        andi   r16,0x0F
        ori    r16,0x30
        mov     r4,r16
        mov     r16,r15
        andi   r16,0x0F
        ori    r16,0x30
        mov     r5,r16

        ret

bBCDx_2:
        ldi     r30,15+1

bBCDx_3:
        ld      r19,-z
        subi   r19,-$03
        sbrc   r19,3
        st     z,r19
        ld      r19,z
        subi   r19,-$30
        sbrc   r19,7
        st     Z,r19
        cpi    ZL,13
        brne  bBCDx_3
        rjmp  bBCDx_1

mul10a:
        swap   r19

mul10b:
        mov     r12,r14
        mov     r13,r15
        lsl     r14
        rol     r15
        lsl     r12
        rol     r13
        lsl     r12
        rol     r13
        lsl     r12
        rol     r13
```

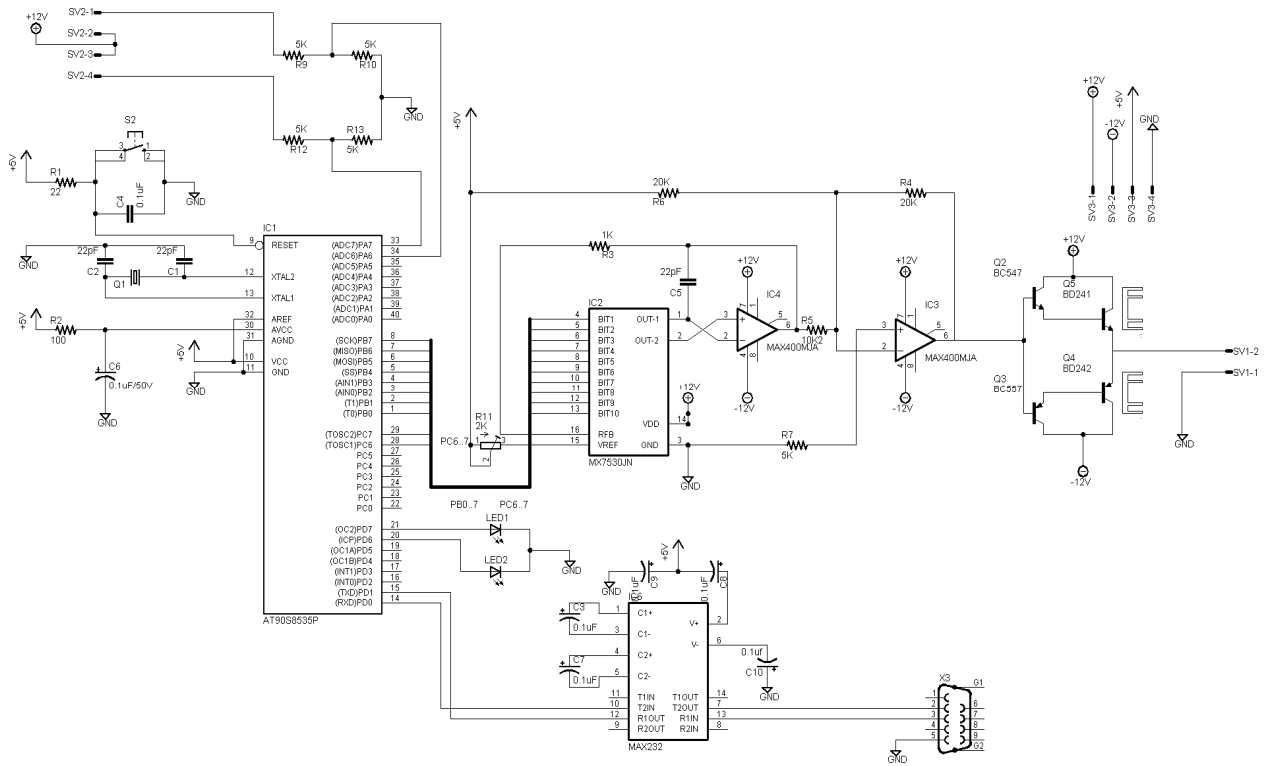
```

add    r14,r12
adc    r15,r13
andi   r19,0x0f
add    r14,r19
brcc   m10_1
inc    r15

m10_1:
ret

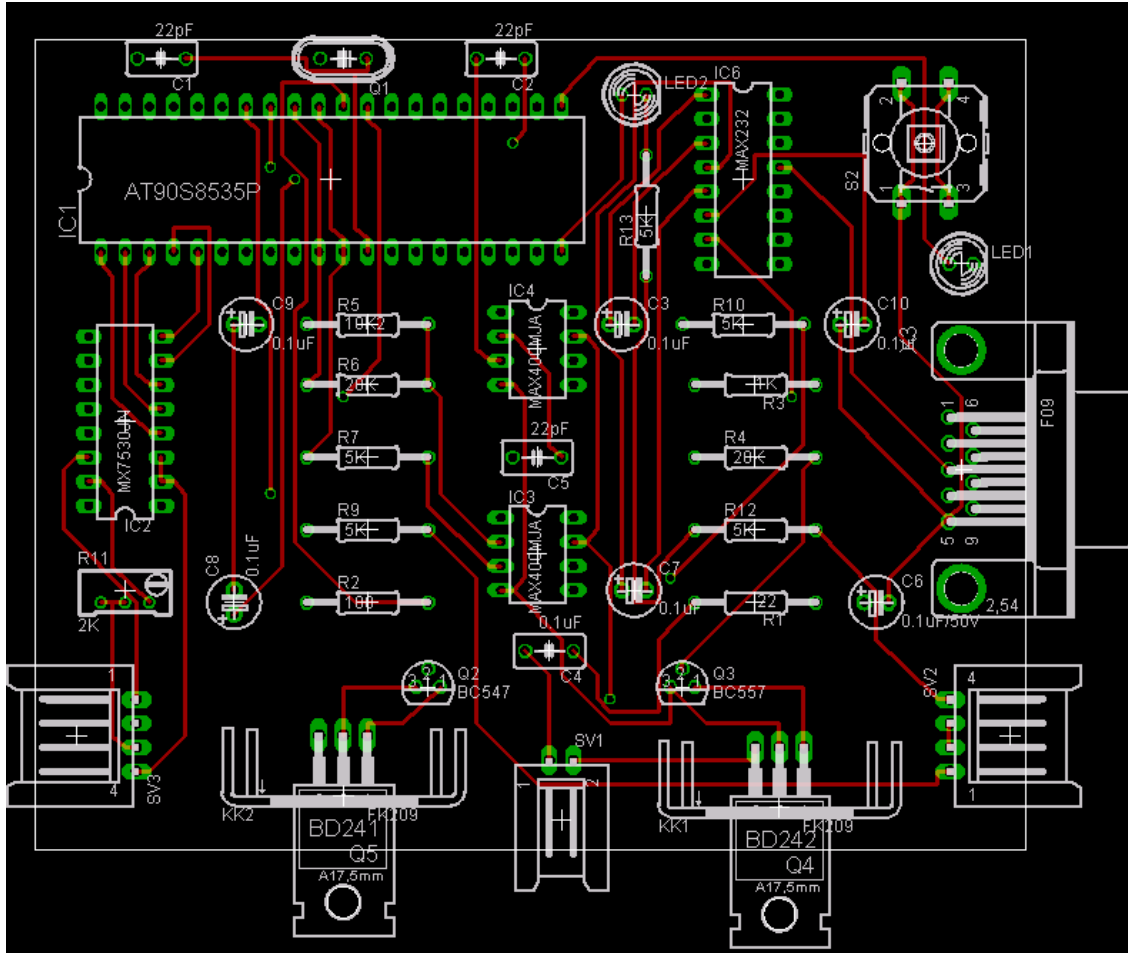
BCD2bin16:
andi   r18,0x0f
clr    r15
mov    r14,r18
mov    r19,r17
rcall  mul10a
mov    r19,r17
rcall  mul10b
mov    r19,r16
rcall  mul10a
mov    r19,r16
rcall  mul10b
ret
    
```

**Skema rangkaian**



## Layout PCB

### Top Layer



Bottom layer

