

BAB III

METODE PENELITIAN

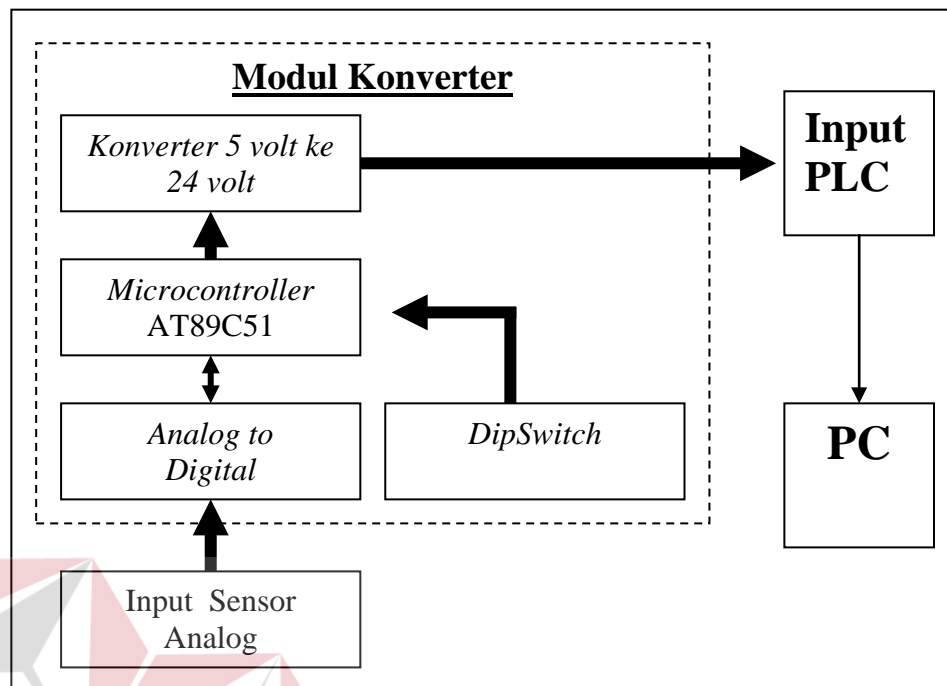
Metode penelitian yang digunakan pada perancangan dan pembuatan perangkat keras dan perangkat lunak yaitu dengan studi kepustakaan dan eksperimen. Dengan cara ini penulis berusaha untuk mendapatkan dan mengumpulkan data-data, informasi, konsep-konsep yang bersifat teoritis dengan membaca buku-buku serta literatur dan bahan-bahan kuliah yang berkaitan dengan permasalahan tersebut. Selanjutnya dilakukan eksperimen untuk pembuatan peralatannya.

Perancangan rangkaian perangkat keras dilakukan setelah memperoleh data-data teoritis permasalahan. Setelah pembuatan perangkat keras, maka dilakukan pengujian terhadap perangkat keras tersebut dengan program-program kecil untuk menguji tiap bagiannya. Terakhir adalah pembuatan perangkat lunak yang sebenarnya sesuai dengan kerja perangkat keras yang diinginkan.

3.1 Perangkat Keras

Pada perancangan dan pembuatan perangkat keras ini, langkah pertama yang harus ditentukan adalah menentukan tujuan dari pembuatan sistem ini. Kemudian merancang blok diagram secara umum atau skematik rangkaian.

Perangkat utama pada alat yang dirancang penulis adalah PLC dan Modul konverter. PLC sebagai modul penerima data, dimana data yang dikirim dari satu *bit input* PLC drubah kembali ke data delapan *bit* dan disimpan ke dalam *register*, dan modul konverter sebagai pengkonversi data delapan *bit* ke satu *bit* yang akan dikirim ke PLC secara *perbit*. Rancangan blok diagram dapat dilihat di bawah ini.



Gambar 3.1. Blok Diagram Sistem Modul Input Analog PLC.

Dari gambar diatas dapat dilihat bahwa perancangan perangkat keras dibagi menjadi beberapa modul, di antaranya : modul minimum sistem *microcontroller* AT89C51, modul ADC, modul penguat tegangan, modul *input* (menggunakan sensor *analog*), dan PLC untuk penyimpanan data dari data sensor.

3.1.1 Minimum Sistem AT89C51

A. Perancangan Minimum Sistem AT89C51

Minimum sistem AT89C51 mempunyai 4Kbyte *Flash* PEROM (*Programmable and Erasable Read Only Memory*), yaitu ROM yang dapat ditulis ulang atau dihapus menggunakan sebuah perangkat *programmer*. *Port 0* digunakan sebagai *bus* data dan 8 *bit bus* alamat rendah dengan sistem *multiplexing addressing* (pengalamatan bergantian). *Port 2* digunakan sebagai 8

bit bus alamat tinggi. *Port 3* dari AT89C51 memiliki fungsi-fungsi khusus sebagai *bus* kontrol antara lain RXD, TXD, $\overline{INT0}$, $\overline{INT1}$, T0, T1, \overline{WR} , dan \overline{RD} .

Microcontroller AT89C51 memiliki osilator *internal* (*on-chip isolator*) yang dapat digunakan sebagai sumber pewaktuan bagi CPU. Dalam perancangan tugas akhir ini menggunakan osilator *internal* tersebut diperlukan sebuah kristal antara *pin* XTAL1 dan *pin* XTAL2 dan sebuah kapasitor ke *ground*.

Periode waktu *timer* secara umum ditentukan oleh persamaan berikut :

Jika kita menggunakan kristal 12 MHz, maka :

$$\begin{aligned}
 \text{Clock} &= 1 / ((1/12) * \text{kristal}) && (3.1) \\
 &= 1 / ((1/12) * 12 \cdot 10^6) \\
 &= 1 / 10^6 \\
 &= 10^{-6} \text{ S} = 1 \text{ uS}
 \end{aligned}$$

B. Pembuatan Minimum Sistem AT89C51

Pin X1 dan X2 dihubungkan ke rangkaian osilator sebagai penentu kecepatan proses dari AT89C51. Program yang ada pada *Flash* PEROM akan dijalankan jika pada saat sistem *direset*, *pin* EA/VP berlogika satu sehingga *microcontroller* aktif berdasarkan program yang ada pada *Flash* PEROM-nya sehingga *pin* EA/VPP dihubungkan dengan VCC +5 volt. *Pin* RESET berfungsi untuk mengembalikan ke keadaan awal proses dari AT89C51. Rangkaian minimum sistem AT89C51 dapat dilihat pada gambar berikut :

ADC0808 yang dihubungkan pada *port 3.2 microcontroller AT89C51*. Program dapat dilihat pada lampiran *listing program microcontroller*.

3.1.2 Modul ADC0808

A. Perancangan Modul ADC0808

Pengubah *analog* ke *digital* atau ADC, adalah alat yang berfungsi untuk mengubah keluaran sinyal *analog* dari sensor ke sinyal *digital*. ADC tipe ini mempunyai *input 8 channel multiplexer* dengan *address logic*, ADC0808 ini digunakan untuk mengkonversi tegangan *analog* (0 sampai 5 V) menjadi data *digital 8 bit*.

Tabel 3.1. Pengalamatan *Input* pada ADC0808.

<i>Selected Analog Channel</i>	<i>Address Line</i>		
	C	B	A
<i>Input 0</i>	L	L	L
<i>Input 1</i>	L	L	H
<i>Input 2</i>	L	H	L
<i>Input 3</i>	L	H	H
<i>Input 4</i>	H	L	L
<i>Input 5</i>	H	L	H
<i>Input 6</i>	H	H	L
<i>Input 7</i>	H	H	H

B. Pembuatan Modul ADC0808

Berikut ini digambarkan hubungan kaki-kaki IC ADC0808 yang dihubungkan dalam pemenuhan kebutuhan sinyal *digital* sebagai masukan pada *port 0 microcontroller AT89C51*.

keluaran ADC. Dikarenakan keluaran dari ADC adalah biner 8 *bit* dengan desimal 0-255 maka konversi dari keluaran ADC ke tegangan adalah (Mazidi, 2000 : 245).

$$V_{in} = \frac{\text{desimal}}{255} \times V_{ref} \quad (3.2)$$

Untuk keluaran ADC (*biner 8 bit*) dalam desimal digunakan persamaan :

$$\text{Desimal} = \left(\frac{V_{in}}{V_{ref}} \right) \times 255 \quad (3.3)$$

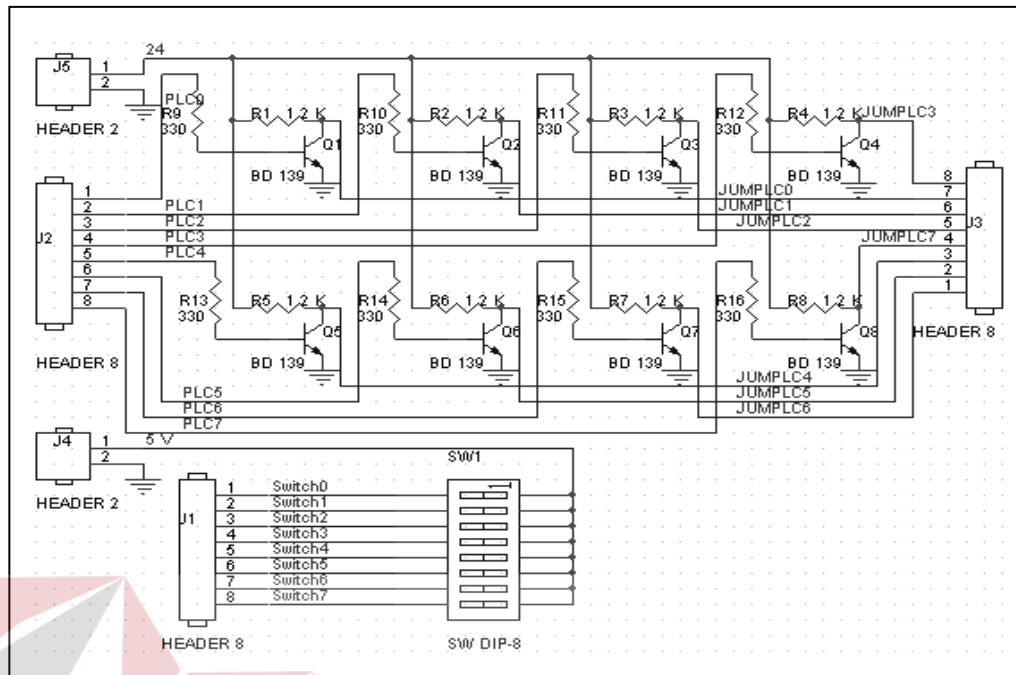
3.1.3 Modul Penguat Tegangan

A. Perancangan Modul Penguat Tegangan

PLC dan *microcontroller* saling berkomunikasi satu arah, dan untuk berkomunikasi antara PLC dengan *microcontroller* tidak bisa langsung dihubungkan karena *output microcontroller* bertegangan 5 volt, sedangkan PLC membutuhkan 11 volt sampai 30 volt agar dapat berlogika 1, untuk itu dibutuhkan penguat tegangan, penguat tegangan disini penulis menggunakan transistor sebagai *switch*.

B. Pembuatan Modul Penguat Tegangan

Berikut ini digambarkan hubungan kaki-kaki IC transistor, pada kaki *basis* transistor dihubungkan ke port 2 *microcontroller*, pada kaki *colector* dihubungkan ke *input* PLC sebagai logika 1 “*high*” dan 0 “*low*”. sedangkan kaki *emitor* masuk ke *ground*.



Gambar 3.4. Skematik Penguat Tegangan dan Tombol Pilih.

C. Cara Kerja Modul Penguat Tegangan

Dari gambar diatas terlihat terdapat transistor BD 139, yang berfungsi menguatkan tegangan dari 5 volt ke 14.8 volt yang dapat diterima oleh PLC sebagai logika 1 “high”. Jika pada *microcontroller* mengeluarkan sinyal *high* di kaki *basis* transistor maka pada kaki *colector* transistor mengeluarkan sinyal *low*, begitu juga sebaliknya jika *microcontroller* mengeluarkan sinyal *low* di kaki *basis* transistor maka pada kaki *colector* transistor mengeluarkan sinyal *high*.

3.1.4 Modul Sensor Analog

Karena Pada Tugas Akhir yang kami buat ini diharapkan dapat digunakan untuk mengontrol sebuah *output* dari *input analog* dengan menggunakan berbagai macam jenis sensor *analog* dengan prinsip kerja dari masing-masing sensor

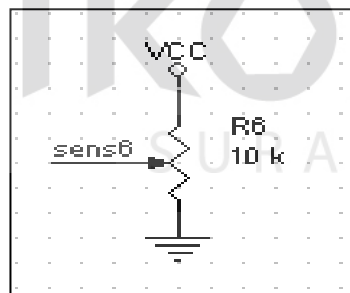
berbeda, maka sensor yang digunakan adalah LM 35, NTC, PTC, LDR (warna), Fotocell, Fotodiode, LDR (ketinggian), Potensiometer.

A.1 Perancangan Modul Sensor Potensiometer

Dalam perancangan potensiometer digunakan potensiometer putar sebagai *input analog* untuk PLC, Potensiometer disebut juga resistor variabel atau disingkat VR (*variable Resistor*). Potensiometer yang dipakai dalam tugas akhir ini digunakan potensiometer ber – CT (*Center Tap* = cabang tengah) adalah potensiometer yang mempunyai 1 kaki tambahan yang bila diukur terhadap kaki utama nilai resistansinya separuh nilai resistansi total (keterangan selengkapnya dapat dilihat pada bab II).

A.2 Pembuatan Modul Sensor Potensiometer

Sensor Potensiometer digunakan sebagai *input analog* bagi PLC.



Gambar 3.5. Rangkaian Elektronik Potensio.

A.3 Cara Kerja Modul Sensor Potensiometer

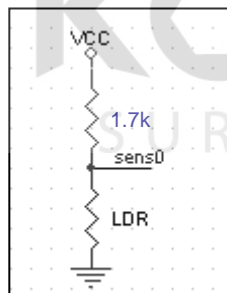
Cara kerja dari sensor ini hanya tinggal memutar tuas dari sensor. Tegangan akan naik jika potensiometer diputar ke arah *ground* dan tegangan akan turun jika potensiometer diputar ke arah VCC (5 volt).

B.1 Perancangan Modul Sensor LDR (warna)

LDR memiliki sensitifitas yang tinggi dengan *time respon* yang cukup baik. Perubahan sensor LDR terpengaruh pada intensitas cahaya semakin terang cahaya yang diterima maka perubahan resistansi yang dihasilkan besar, dan sebaliknya ketika tidak adanya cahaya yang masuk atau intensitas cahaya rendah maka resistansi yang dihasilkan juga rendah. Dari karakteristik ini bisa kita ketahui bahwa nilai resistansi yang dihasilkan berbanding lurus dengan cahaya yang masuk, semakin tinggi intensitas cahaya yang masuk maka nilai resistansi yang dihasilkan semakin naik.

B.2 Pembuatan Modul Sensor LDR (warna)

Sensor LDR digunakan untuk mendeteksi warna dari benda dengan membedakan warna dari benda itu sendiri. warna yang dideteksi diantaranya warna putih, warna hitam, warna merah, dan warna silver.



Gambar 3.6. Rangkaian Elektronik LDR (warna).

B.3 Cara Kerja Modul Sensor LDR (warna)

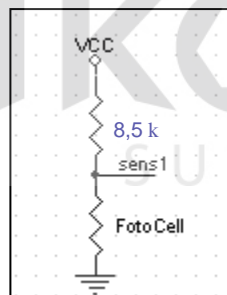
Sensor LDR bekerja dengan perbedaan intensitas cahaya yang diterima dari pantulan benda yang berwarna, semakin terang warna benda maka menghasilkan tegangan yang lebih besar dibandingkan dengan warna gelap.

C.1 Perancangan Modul Sensor Fotocell

Fotocell memiliki sensitifitas yang tinggi dengan *time respon* yang cukup baik. Perubahan sensor Fotocell terpengaruh pada intensitas cahaya semakin terang cahaya yang diterima maka perubahan resistansi yang dihasilkan besar, dan sebaliknya ketika tidak adanya cahaya yang masuk atau intensitas cahaya rendah maka resistansi yang dihasilkan juga rendah. Dari karakteristik ini bisa kita ketahui bahwa nilai resistansi yang dihasilkan berbanding lurus dengan cahaya yang masuk, semakin tinggi intensitas cahaya yang masuk maka nilai resistansi yang dihasilkan semakin naik.

C.2 Pembuatan Modul Sensor Fotocell

Sensor Fotocell digunakan untuk mendeteksi warna dari benda dengan membedakan warna dari benda itu sendiri. warna yang dideteksi diantaranya warna putih, warna hitam, warna merah, dan warna silver.



Gambar 3.7. Rangkaian Elektronik Fotocell.

C.3 Cara Kerja Modul Sensor Fotocell

Sensor Fotocell bekerja dengan perbedaan intensitas cahaya yang diterima dari pantulan benda yang berwarna, semakin terang warna benda maka menghasilkan tegangan yang lebih besar dibandingkan dengan warna gelap.

D.1 Perancangan Modul Sensor NTC

NTC memiliki sensitifitas yang tinggi dengan *time respon* yang cukup baik. Ketika ia berada pada temperatur 32°C maka ia menghasilkan nilai resistansi 848 Ω sampai 863 Ω, sedangkan pada temperatur 39°C NTC menghasilkan nilai resistansi sebesar 653 Ω sampai 668 Ω. Dari karakteristik ini bisa kita ketahui bahwa nilai resistansi yang dihasilkan tidak berbanding lurus dengan temperatur yang diukur, semakin tinggi temperatur yang diukur maka nilai resistansi yang dihasilkan semakin turun.

Output sensor NTC adalah resistansi sedangkan *output* yang dapat diterima oleh ADC0808 adalah tegangan sehingga diperlukan pengkonversian ke tegangan dengan jalan membuat rangkaian pembagi tegangan. Rangkaian ini diharapkan memiliki *range output* yang lebar. Sensor NTC difungsikan sebagai variabel resistor dan ditambah dengan sebuah resistor sebagai pembagi tegangan.

Range output NTC untuk temperatur 32 °C sampai 39 °C adalah 863 Ω sampai dengan 668 Ω sehingga sebagai pembagi tegangan digunakan resistor dengan nilai 1 KΩ. Sebagai tegangan sumber digunakan tegangan sebesar 5 Volt. Perhitungan besarnya *output* yang dihasilkan berdasarkan persamaan berikut.

- a. Temperatur 32°C

$$V_D = \frac{R_2 V_s}{R_1 + R_2} \quad (3.4)$$

$$V_D = \frac{863}{1000 + 863} \times 5V$$

$$= 2,31 \text{ Volt}$$

b. Temperatur 39°C

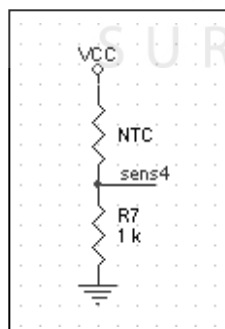
$$V_D = \frac{668}{1000 + 668} \times 5\text{v}$$

$$= 2,00 \text{ Volt}$$

Dengan *range* tegangan untuk temperatur 32° C sampai 39° C yang cukup lebar.

D.2 Pembuatan Modul Sensor NTC

Sensor ini digunakan untuk mengukur suhu dengan cara membedakan suhu ruangan. Bersamaan dengan perubahan suhu pada ruangan maka suhu yang diukur juga mengalami perubahan dan mengakibatkan tegangan yang dihasilkan juga berubah. Jika suhu yang diukur mengalami kenaikan maka tegangan yang dihasilkan juga mengalami kenaikan. Begitu pula sebaliknya jika suhu mengalami penurunan maka tegangan yang dihasilkan mengalami penurunan.



Gambar 3.8. Rangkaian Elektronik NTC.

D.3 Cara Kerja Modul Sensor NTC

Data dari sensor ini akan berubah sesuai dengan perubahan suhu ruang yang diukur. Data ini kemudian akan diterima oleh ADC dan dikirim ke *microcontroller* yang kemudian akan diteruskan ke PLC.

E.1 Perancangan Modul Sensor PTC

PTC memiliki sensitifitas yang tinggi dengan *time respon* yang cukup baik. Ketika ia berada pada temperatur 41°C maka ia menghasilkan nilai resistansi $11.3\ \Omega$ sampai dengan $11.4\ \Omega$, sedangkan pada temperatur 32°C PTC menghasilkan nilai resistansi sebesar $10.3\ \Omega$. Dari karakteristik ini bisa kita ketahui bahwa nilai resistansi yang dihasilkan berbanding lurus dengan temperatur yang diukur, semakin tinggi temperatur yang diukur maka nilai resistansi yang dihasilkan semakin naik.

Output sensor PTC adalah resistansi sedangkan *output* yang dapat diterima oleh ADC0808 adalah tegangan sehingga diperlukan pengkonversian ke tegangan dengan jalan membuat rangkaian pembagi tegangan. Rangkaian ini diharapkan memiliki *range output* yang lebar. Sensor PTC difungsikan sebagai variabel resistor dan ditambah dengan sebuah resistor sebagai pembagi tegangan.

Range output PTC untuk temperatur 32°C sampai 36°C adalah $10,3\ \Omega$ sampai dengan $11.4\ \Omega$ sehingga sebagai pembagi tegangan digunakan resistor dengan nilai $56\ \Omega$. Sebagai tegangan sumber digunakan tegangan sebesar 5Volt . Perhitungan besarnya *output* yang dihasilkan berdasarkan persamaan berikut.

a. Temperatur 32°C

$$V_D = \frac{R_1 V_s}{R_1 + R_2} \quad (3.5)$$

$$V_D = \frac{10.3}{10.3 + 56} \times 5V$$

$$= 0,07 \text{ Volt}$$

b. Temperatur 36° C

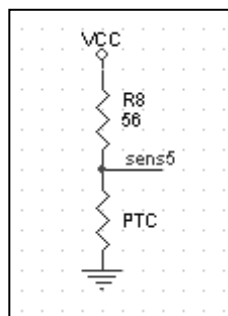
$$V_D = \frac{11.4}{11.4 + 56} \times 5V$$

$$= 0,84 \text{ Volt}$$

Dengan *range* tegangan untuk temperatur 32° C sampai 36° C yang cukup lebar.

E.2 Pembuatan Modul Sensor PTC

Sensor ini digunakan untuk mengukur suhu dengan cara membedakan suhu ruangan. Bersamaan dengan perubahan suhu pada ruangan maka suhu yang diukur juga mengalami perubahan dan mengakibatkan tegangan yang dihasilkan juga berubah. Jika suhu yang diukur mengalami kenaikan maka tegangan yang dihasilkan juga mengalami kenaikan. Begitu pula sebaliknya jika suhu mengalami penurunan maka tegangan yang dihasilkan mengalami penurunan.



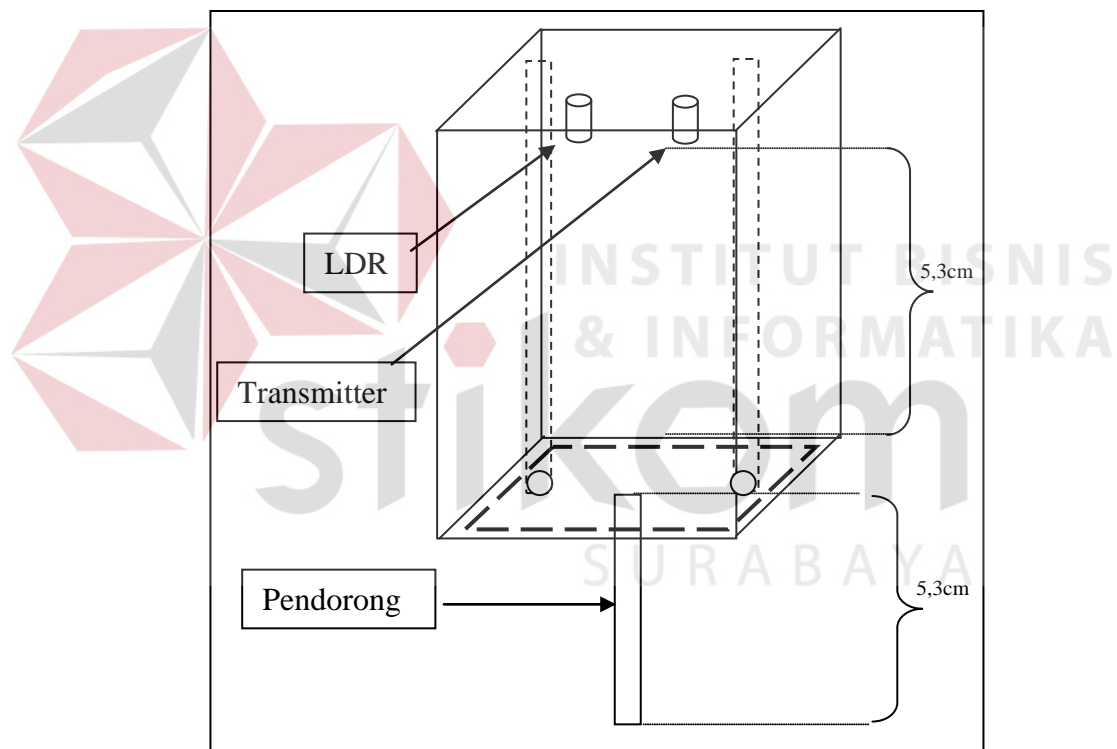
Gambar 3.9. Rangkaian Elektronik PTC.

F.3 Cara Kerja Modul Sensor LM35

Data dari sensor ini akan berubah sesuai dengan perubahan suhu ruang yang diukur. Data ini kemudian akan diterima oleh ADC dan dikirim ke *microcontroller* yang kemudian akan diteruskan ke PLC.

G.1 Perancangan Modul Sensor LDR(ketinggian)

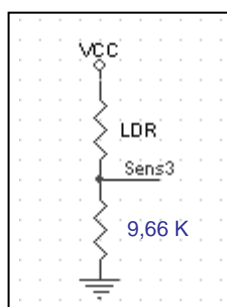
Dalam perancangan LDR kali ini sama dengan LDR (warna), namun dalam perancangan LDR (ketinggian) memanfaatkan lebar sempitnya ruangan.



Gambar 3.11. Kontruksi Sensor LDR (ketinggian).

G.2 Pembuatan Modul Sensor LDR(ketinggian)

Sensor LDR juga dapat difungsikan sebagai pengukur ketinggian dari suatu benda, dengan memanfaatkan lebar sempitnya ruang.



Gambar 3.12. Rangkaian Elektronik LDR (ketinggian).

G.3 Cara Kerja Modul Sensor LDR(ketinggian)

Sensor LDR yang berfungsi sebagai sensor ketinggian bekerja dengan cara dengan metode penyempitan ruang dan pelebaran ruang, sehingga intensitas cahaya yang dipantulkan lebih terang dan menghasilkan tegangan yang besar ketika keadaan ruang sempit dan ketika keadaan ruang lebar atau tidak sempit cahaya yang dipantulkan akan berkurang dan tegangan yang dihasilkan juga berkurang sesuai dengan pelebaran ruang.

H.1 Perancangan Modul Sensor Fotodiode

H.1.1 Transmitter IR

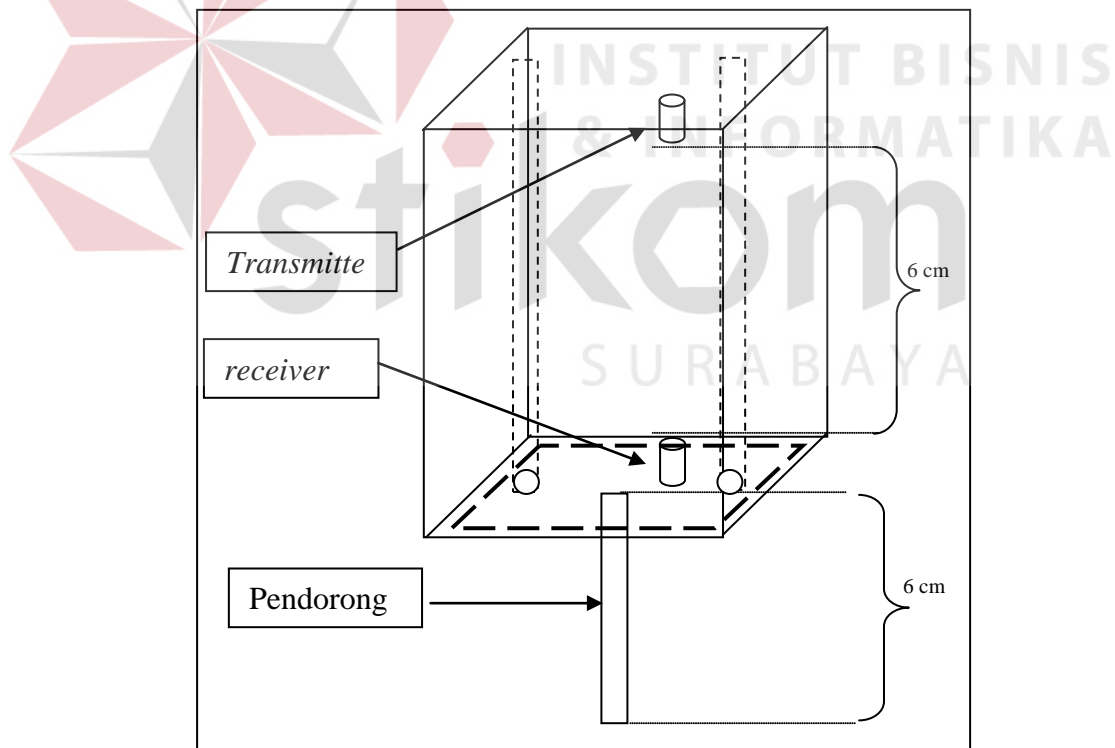
Pada *transmitter* terdapat sebuah resistor dengan nilai 270 yang dirangkai seri dengan sebuah LED. IR (Infra Red). Resistor pada rangkaian tersebut mempunyai fungsi sebagai pembatas arus yang masuk pada LED IR sehingga penggunaan LED IR lebih awet (tidak mudah terbakar). Sedangkan LED IR pada *transmitter* berfungsi sebagai penghasil cahaya infrared.

H.1.2 Receiver IR

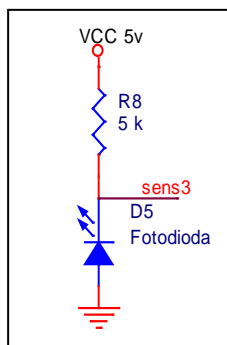
Untuk *receiver* infrared digunakan fotodiode, karena fotodiode dapat menerima cahaya infrared. Dengan teori pembagi tegangan, V_{in} pada receiver IR dapat diatur, dengan maksud agar Rangkaian ini dapat memiliki *range* output yang lebar. sebagai pembagi tegangan digunakan resistor dengan nilai 5 K Ω . Sebagai tegangan sumber digunakan tegangan sebesar 5 Volt.

H.2 Pembuatan Modul Sensor Fotodiode

Sensor infrared dan fotodiode difungsikan sebagai pengukur ketinggian dari suatu benda, dengan memanfaatkan jarak antara infrared (*transmitter*) dengan fotodiode (*receiver*).



Gambar 3.13. Kontruksi Sensor Fotodiode.



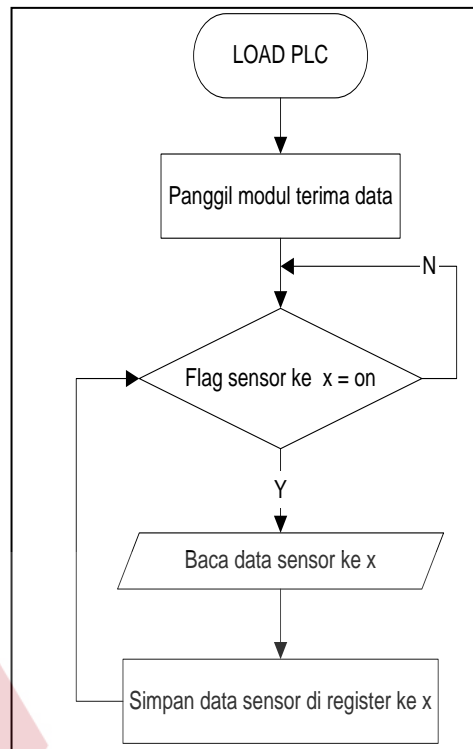
Gambar 3.14. Rangkaian Elektronik Fotodiode.

H.3 Cara Kerja Modul Sensor Fotodiode

Cara kerja dari sensor ini membandingkan jarak dari sinyal yang dikirim *transmitter* (Infrared) dengan penerima (Fotodiode) semakin dekat jarak dari pengirim dan penerima tegangan yang dihasilkan akan naik dan semakin jauh jarak sensor dari pengirim dan penerima tegangan yang dihasilkan juga akan turun.

3.1.5. Perancangan Unit Kendali PLC

Dalam perancangan perangkat keras untuk PLC, penulis hanya menyusun dan merancang modul penerimaan data dari satu *bit input* PLC. Langkah awal penulis harus menggambarkan terlebih dahulu diagram blok dari tugas akhir penulis seperti yang terlihat pada gambar 3.15.



Gambar 3.15. *Flowchart* Modul Terima Data.

A.1 Perancangan *Flowchart* Sistem secara Keseluruhan

Jika dilihat dari gambar diatas, maka langkah awal dari *flowchart* yang dirancang adalah, program PLC di *load* setelah itu panggil modul terima data sebagai tanda dimulainya penerimaan data. Sinyal akan di baca oleh PLC jika terjadi pengaktifan *flag* untuk sensor yang akan dibaca, dan untuk *flag* sensor yang dinonaktifkan tidak akan terjadi pembacaan sensor. Jika terjadi penonaktifan modul terima data menandakan dihentikannya pembacaan data sensor *analog*.

A.2 Perancangan Perangkat I/O PLC

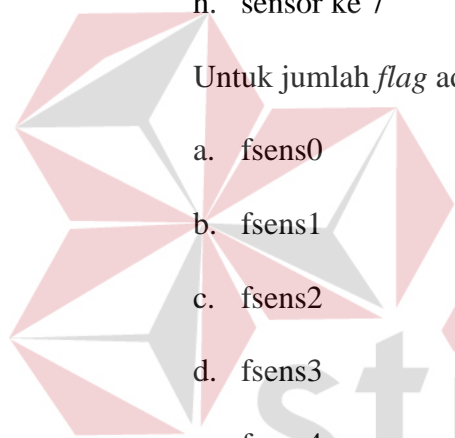
Jika dilihat dari gambar 3.15 maka jumlah *input* PLC yang akan digunakan oleh penulis untuk disain sistem penerimaan data yang akan di rancang adalah sebagai berikut :

Untuk jumlah *input* ada delapan yaitu:

- a. sensor ke 0
- b. sensor ke 1
- c. sensor ke 2
- d. sensor ke 3
- e. sensor ke 4
- f. sensor ke 5
- g. sensor ke 6
- h. sensor ke 7

Untuk jumlah *flag* ada 14 yaitu :

- a. fsens0
- b. fsens1
- c. fsens2
- d. fsens3
- e. fsens4
- f. fsens5
- g. fsens6
- h. fsens7
- i. flag0
- j. flag1
- k. flag2
- l. flag3
- m. flag4
- n. flag5
- o. flag6



INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA

p. flag7

Untuk jumlah *register* ada sepuluh yaitu:

- a. R0 untuk data sensor ke 0
- b. R1 untuk data sensor ke 1
- c. R2 untuk data sensor ke 2
- d. R3 untuk data sensor ke 3
- e. R4 untuk data sensor ke 4
- f. R5 untuk data sensor ke 5
- g. R6 untuk data sensor ke 6
- h. R7 untuk data sensor ke 7
- i. R8 untuk terima data 0 dan 1
- j. R9 untuk penyimpanan data sementara

A.3 Perancangan *Allocation List* dan *Address* pada PLC

Allocation list adalah daftar yang berisi pemetaan sensor alamat *input* PLC, dengan adanya *allocation list* ini sangat membantu dalam mengerjakan program dalam PLC, seperti yang terlihat pada tabel dibawah ini:

Tabel 3.2. *Allocation List* untuk *Input*.

NO	Sensor / Aktuator	Nama Simbolis	Alamat PLC
1	Sensor 0	Inx	I0.0
2	Sensor 1	Inx1	I0.1
3	Sensor 2	Inx2	I0.2
4	Sensor 3	Inx3	I0.3

5	Sensor 4	Inx4	I0.4
6	Sensor 5	Inx5	I0.5
7	Sensor 6	Inx6	I0.6
8	Sensor 7	Inx7	I0.7

Tabel 3.3. *Allocation List* untuk *Flag*.

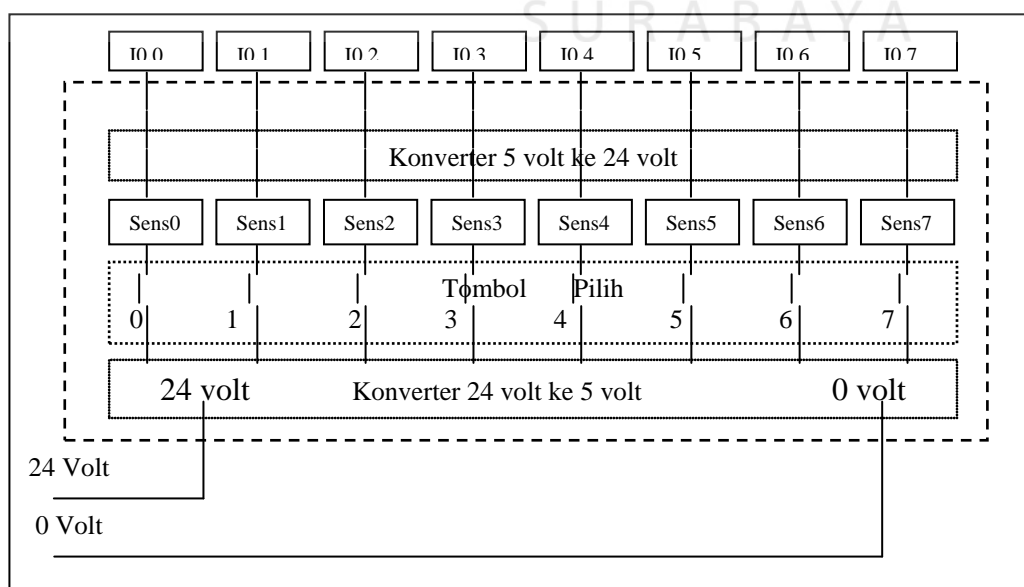
NO	Sensor / Aktuator	Nama Simbolis	Alamat PLC	Keterangan Sinyal
1	Sensor 0	Fsens0	F0.0	0 = data sensor 0 tidak dibaca 1 = data sensor 0 dibaca
2	Sensor 1	Fsens1	F0.1	0 = data sensor 1 tidak dibaca 1 = data sensor 1 dibaca
3	Sensor 2	Fsens2	F0.2	0 = data sensor 2 tidak dibaca 1 = data sensor 2 dibaca
4	Sensor 3	Fsens3	F0.3	0 = data sensor 3 tidak dibaca 1 = data sensor 3 dibaca
5	Sensor 4	Fsens4	F0.4	0 = data sensor 4 tidak dibaca 1 = data sensor 4 dibaca
6	Sensor 5	Fsens5	F0.5	0 = data sensor 5 tidak dibaca 1 = data sensor 5 dibaca
7	Sensor 6	Fsens6	F0.6	0 = data sensor 6 tidak dibaca 1 = data sensor 6 dibaca
8	Sensor 7	Fsens7	F0.7	0 = data sensor 7 tidak dibaca 1 = data sensor 7 dibaca
9	Sensor 0	Flag0	F1.0	0 = data sensor 0 tidak dibaca 1 = data sensor 0 dibaca
10	Sensor 1	Flag1	F1.1	0 = data sensor 1 tidak dibaca 1 = data sensor 1 dibaca
11	Sensor 2	Flag2	F1.2	0 = data sensor 2 tidak dibaca 1 = data sensor 2 dibaca
12	Sensor 3	Flag3	F1.3	0 = data sensor 3 tidak dibaca 1 = data sensor 3 dibaca
13	Sensor 4	Flag4	F1.4	0 = data sensor 4 tidak dibaca 1 = data sensor 4 dibaca
14	Sensor 5	Flag5	F1.5	0 = data sensor 5 tidak dibaca 1 = data sensor 5 dibaca
15	Sensor 6	Flag6	F1.6	0 = data sensor 6 tidak dibaca 1 = data sensor 6 dibaca
16	Sensor 7	Flag7	F1.7	0 = data sensor 7 tidak dibaca 1 = data sensor 7 dibaca

Tabel 3.4. *Allocation List* untuk Register.

NO	Sensor / Aktuator	Nama Simbolis	Keterangan Sinyal
1	Sensor 0	R0	data sensor 0
2	Sensor 1	R1	data sensor 1
3	Sensor 2	R2	data sensor 2
4	Sensor 3	R3	data sensor 3
5	Sensor 4	R4	data sensor 4
6	Sensor 5	R5	data sensor 5
7	Sensor 6	R6	data sensor 6
8	Sensor 7	R7	data sensor 7
9	Sensor 0	R8	0 = data bernilai 0 1 = data bernilai 1
10	Sensor 1	R9	data sensor ke X untuk sementara

A.4 Perancangan Diagram Rangkaian Listrik.

Diagram rangkaian listrik merupakan diagram yang menunjukkan hubungan antara catu daya, PLC, dan sensor. Diagram ini sangat berguna sebagai panduan pemasangan peralatan agar tidak terjadi hubungan singkat yang bisa merusak peralatan, seperti yang terlihat pada gambar di bawah ini.



Gambar 3.16. Diagram Rangkaian Listrik.

3.2 Perancangan Perangkat Lunak

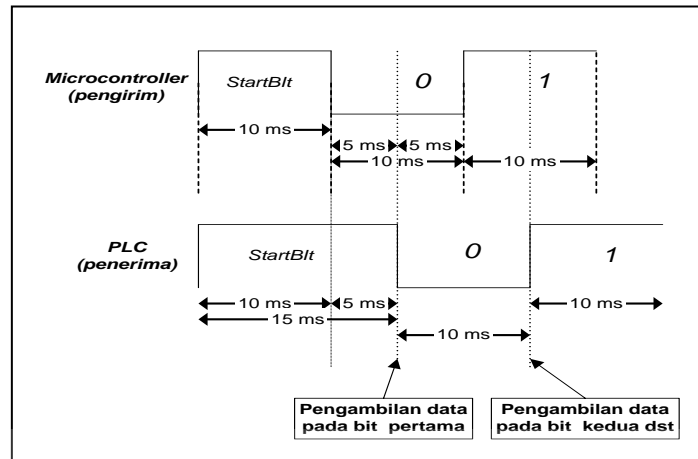
Untuk perancangan perangkat lunak, yaitu perangkat lunak yang digunakan untuk mengendalikan *hardware*. Dalam perancangan perangkat lunak terdapat dua macam bahasa pemrograman yang digunakan oleh penulis. Yang pertama bahasa pemrograman STL yang merupakan bahasa pemrograman dari FESTO yang dirancang khusus untuk PLC FESTO. Bahasa pemrograman yang kedua yaitu bahasa pemrograman yang digunakan untuk *microcontroller* 89C51 dengan menggunakan Bahasa *Ansi C*. Semua Bahasa *Ansi C* ditulis dengan menggunakan program *Franklin Software (Proview32)*, Bahasa *Ansi C* yaitu bahasa tingkat tinggi, yang selanjutnya harus menggunakan bantuan rangkaian *writer* untuk mengisi *microcontroller* tersebut.

A. Perancangan Protokol Komunikasi Data

Sebelum membuat program STL dan *Ansi C*, sebaiknya merancang protokol komunikasi data antara PLC dan *microcontroller*. *Microcontroller* mengirim data secara *serial* sebanyak 8 *bit* lewat jalur *output microcontroller*. Protokol komunikasi data menggunakan metode asinkron, sehingga dibutuhkan *start bit* dan *stop bit*, dan diterima oleh sebuah *input* pada PLC.

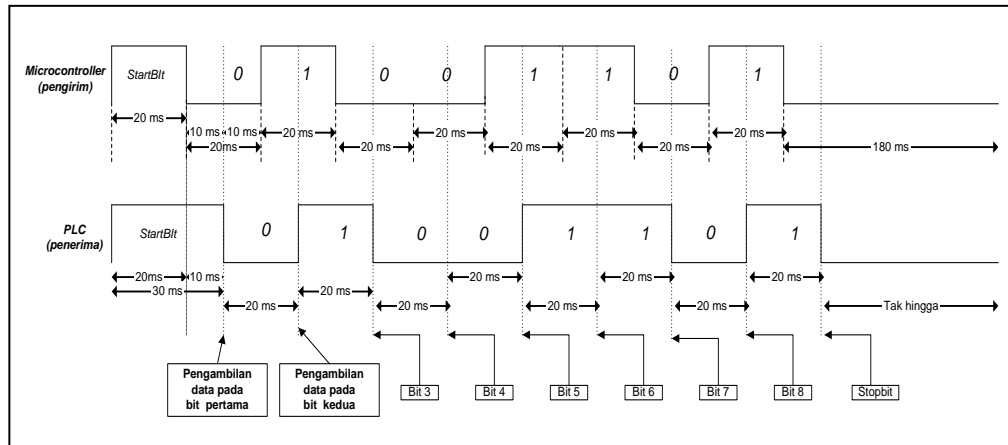
Pada perancangan pertama, diharapkan PLC sebagai penerima data dapat membaca data yang dikirim oleh *microcontroller* secara cepat, dalam pengambilan data dari *microcontroller* dan untuk membedakan data pada *bit* pertama dengan yang berikutnya (sampai *bit* ke delapan) digunakan *timer* yang sudah ada dalam PLC. *Timer* pada PLC kelipatan 10 ms. Dalam perancangan pertama pada *microcontroller* untuk *start bit*, dan data (tiap *bit*) di beri *interval*

10ms, sedangkan pada sisi penerima (PLC) untuk *start bit* 15 ms, dan untuk penerimaan data tiap *bit* 10 ms. Diagram rancangan dapat dilihat pada gambar dibawah ini.



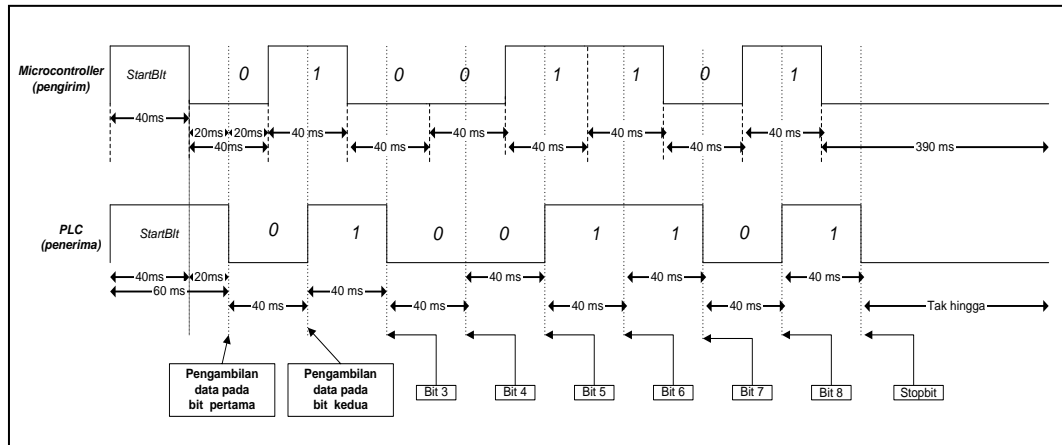
Gambar 3.17. Protokol Komunikasi 1.

Dalam diagram diatas dapat dilihat pada PLC terdapat *start bit* yang bernilai 15 ms yang dimaksudkan agar pengambilan data tepat di tengah-tengah saat pengiriman data terjadi, Permasalahan yang terjadi pada *timer* PLC tidak dapat di beri nilai 15 ms, karena nilai *timer* pada PLC harus kelipatan 10 ms, maka dalam rancangan diatas tidak dapat dipakai sehingga pada sisi pengirim harus di sesuaikan dengan kondisi pada sisi penerima. Untuk perancangan yang kedua pada sisi penerima (PLC) untuk *start bit* 30 ms , dan (tiap *bit*) 20 ms, untuk *stop bit* pada PLC menunggu sinyal low(0), sedangkan pada sisi pengirim (*microcontroller*) untuk *start bit*, dan (tiap *bit*) 20 ms, untuk *stop bit* didapatkan dari penjumlahan *timing clock* pada 1 *frame* di sisi penerima ditambah 10 ms sehingga didapatkan 180 ms, Diagram rancangan dapat dilihat pada gambar di bawah ini.



Gambar 3.18. Protokol Komunikasi 2.

Dalam perancangan protokol komunikasi yang kedua data dapat di terima dengan baik oleh PLC untuk satu data, dalam perancangan yang kedua terdapat kelemahan saat terjadi pengiriman delapan data secara langsung oleh *microcontroller*, PLC tidak dapat menerima data secara baik sehingga perancangan yang kedua tidak dapat dipakai dalam penerimaan 8 data secara langsung. Maka dirancang kembali dengan menambahkan waktu tunda pada penerimaan dan pengiriman data, didapatkan pada sisi penerima (PLC) untuk *start bit* 60 ms, dan (tiap *bit*) 40 ms, untuk *stop bit* pada PLC menunggu sinyal *low(0)*, sedangkan pada sisi pengirim (*microcontroller*) untuk *start bit*, dan data (tiap *bit*) 40 ms, untuk *stop bit* didapatkan dari penjumlahan *timing clock* pada 1 *frame* di sisi penerima ditambah 10 ms sehingga didapatkan 390 ms, Diagram rancangan dapat dilihat pada gambar di bawah ini.

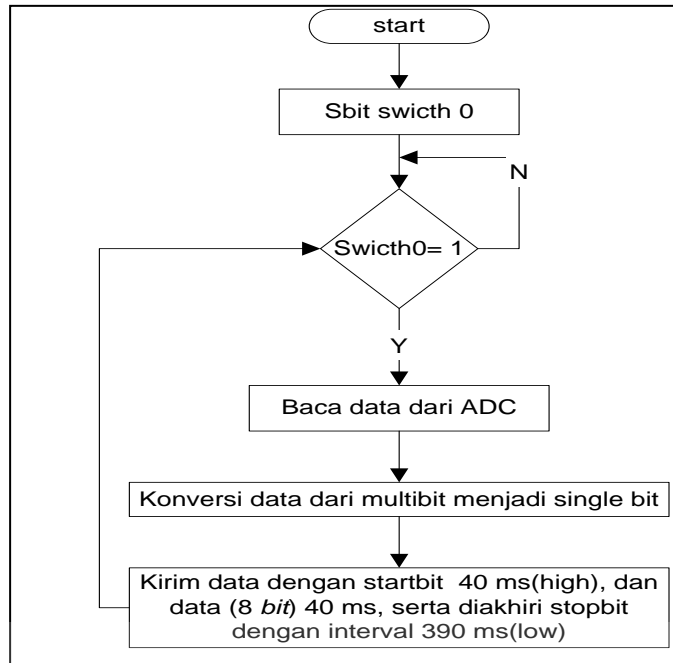


Gambar 3.19. Protokol Komunikasi 3.

Pada perancangan yang ketiga didapatkan hasil yang diinginkan yaitu PLC dapat menerima 8 data secara langsung dengan baik yang dikirim oleh microcontroller.

B. Perancangan Komunikasi Data pada *Microcontroller*

Perancangan untuk komunikasi data yang akan dibuat yaitu mencoba untuk mengirim sebuah data sensor *analog* oleh *microcontroller* ke PLC. Data dikirim sesudah *start bit* dan sebelum *stop bit*, untuk lebih jelasnya dapat dilihat pada *flowchart* dibawah ini .



Gambar 3.20. *FlowChart* Program Kirim data dari *Microcontroller*.

Untuk memperjelas keterangan dari *flowchart* diatas. Disini akan diperlihatkan *listing* program komunikasi pengiriman data *microcontroller* ke PLC untuk sebuah data *analog*.

```

// pengiriman satu data dari ADC -> Micro -> PLC

#include <reg51.h>
#include <stdio.h>
#include <absacc.h>
#include <string.h>

#define SensKn XBYTE[0x0000] //address

sbit Switch0=P1^0; //pin 10
sbit data0=P2^0; //pin 20
sbit eoc=P3^2; //untuk menrima intruksi pembacaan data pada adc

void delay(int);
void Baca();
void check();
void DecToBin();
void Kirim();

int data_array[8];
int DtSens;

void main()
{
    P2=0xff;
    delay(38); //delay yang seharusnya dihasilkan 380 ms
    while(1)
    {
        if (switch0=1)
        {
            Baca();
            DecToBin();
            Kirim();
        }
    }
}

```

```

    }
}

void Kirim()
{
    int i;
    data0=0;
    delay(3);
    for (i=7;i>=0;i--)
    {
        if (data_array[i]==0)
            { data0=0; }
        else
            { data0=1; }
        delay(3);
    }
    data0=1;
    delay(38);
}

void DecToBin()
{
    int i,bantu,bantul;           //variabel bebas
    bantu=DtSens;
    for(i=0;i<8;i++)
    {
        bantu=bantu/2;
        if ((bantu%2)==0)
            {data_array[i]=0;}
        else
            {data_array[i]=1;}
        bantu=bantul;
    }
}

void check()
{
    while(!eoc);
    while(eoc);
}

void Baca()
{
    SensKn=0x00;
    check();
    DtSens=SensKn;
}

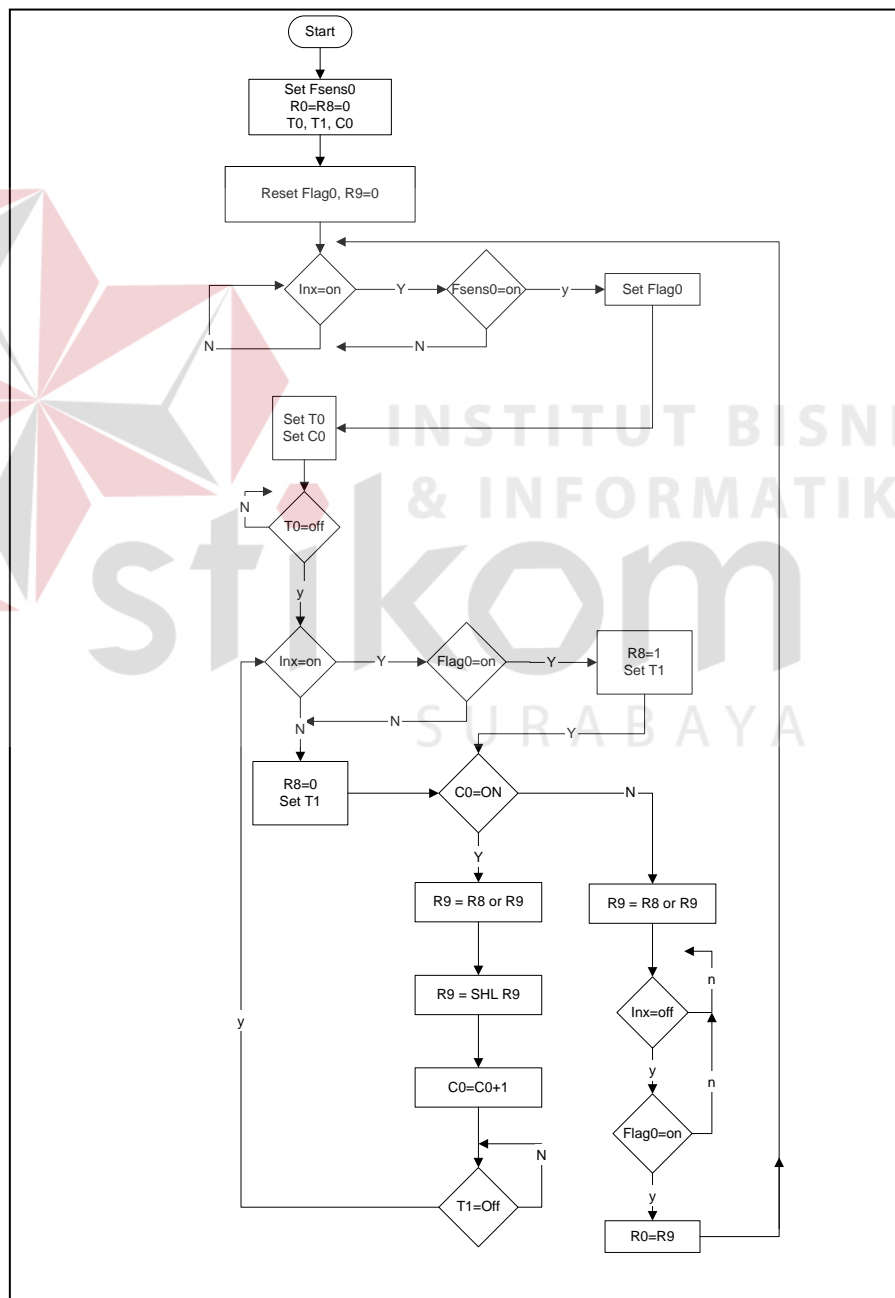
void delay(int lama)
{
    int i;
    TMOD =0x01;
    for(i=0;i<=lama;i++)
    {
        TH0=0xd8;
        TL0=0xef;
        TR0=1;
        while(!TF0);
        TR0=0;
        TF0=0;
    }
}

```

Karena yang dikirim terdapat 8 kanal sensor analog program diatas di *looping* sebanyak delapan kali .

C. Perancangan Komunikasi Data pada PLC

Perancangan untuk komunikasi data PLC yang akan dibuat yaitu mencoba untuk menerima sebuah data sensor *analog* yang dikirim oleh *microcontroller*. Data diterima sesudah *start bit* dan sebelum *stop bit*, untuk lebih jelasnya dapat dilihat pada *flowchart* dibawah ini .



Gambar 3.21. Flow Chart Program Terima Data dari PLC.

Untuk memperjelas *flowchart* diatas, dibawah ini dilampirkan sebuah

listing program PLC untuk terima satu data dari *microcontroller*.

```

0001 STEP init (1)
0002 THEN LOAD V6 "delay startbit
0003 TO TP0
0004 LOAD V4 "delay data 0-7 & stopbit
0005 TO TP1
0006 LOAD V7 "looping data 0-7 (biner -> `
"desimal)
0007 TO CP0
0008 LOAD V0 "penyimpanan data
0009 TO R0 "sens0
0010 TO R8 "untuk penerimaan data
" (0/1)

0011
0012 STEP nol (2)
0013 THEN LOAD V0
0014 TO R9 "untuk perhitungan
"digital to desimal

0015 RESET flag0
0016
0017 STEP Cek_Start (3)
0018 IF (
0019 AND (
0020 THEN SET fsens0 ) "sens0
flag0
0021 JMP TO set_bantu (4)
0022

0023 STEP set_bantu (4)
0024 THEN SET T0
0025 SET C0
0026
0027 STEP startbit (5) "startbit aktif high
0028 IF N T0
0029 THEN NOP
0030
0031STEP BacaData (6)
0032 IF (
0033 AND (
0034 THEN JMP TO baca_satu (7)
0035 OTHRW JMP TO baca_nol (8)
0036

0037 STEP baca_satu (7)
0038 THEN LOAD V1
0039 TO R8
0040 SET T1
0041 JMP TO Bin2Dec (9)
0042

0043 STEP baca_nol (8)
0044 THEN LOAD V0
0045 TO R8
0046 SET T1
0047

0048 STEP Bin2Dec (9)
0049 IF C0
0050 THEN LOAD ( R8
0051 OR R9 )
0052 TO R9
0053 LOAD R9 "geser ke kiri
0054 SHL
0055 TO R9
0056 INC C0
0057 JMP TO cek (11)
0058

0059 OTHRW LOAD ( R8
0060 OR R9 )
0061 TO R9
0062

0063 STEP stopbit (10) "stopbit aktif low (0)
0064 IF ( N inx

```

```
0065         AND          flag0      )
0066 THEN LOAD          R9
0067         TO           R0
0068         JMP TO      nol          (2)
0069
0070 STEP cek          (11)
0071 IF              N      T1
0072 THEN JMP TO      BacaData      (6)
```

Sebagaimana *microcontroller* yang dikirim adalah sebanyak delapan kanal maka program PLC diatas juga *looping* sebanyak delapan.

