

SKRIPSI

PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI MULTI INKUBATOR MENGGUNAKAN PLC *SMART RELAY* *ZELIO LOGIC* BERBASIS SCADA



Disusun Oleh :

**PUTRA HADI PRAWIRA
05.12.011**

**KONSENTRASI TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
SEPTEMBER 2010**

12/11/82

RESEARCH AND DEVELOPMENT DIVISION
GENERAL INVESTIGATIVE DIVISION
FEDERAL BUREAU OF INVESTIGATION
UNITED STATES DEPARTMENT OF JUSTICE

RECEIVED

NOV 23 1982
FBI - MEMPHIS

MEMPHIS TELETYPE UNIT
NOV 23 1982
FBI - MEMPHIS
COMMUNICATIONS SECTION

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI MULTI INKUBATOR MENGGUNAKAN PLC *SMART RELAY* *ZELIO LOGIC* BERBASIS SCADA

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelara Sarjana Teknik Energi Listrik Strata Satu (S-1)*

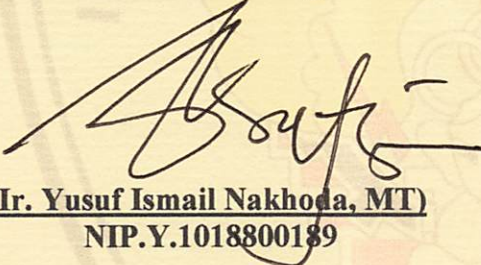
Disusun Oleh :

PUTRA HADI PRAWIRA
NIM : 05.12.011

Diperiksa dan Disetujui,

Dosen Pembimbing 1

Dosen Pembimbing 2


(Ir. Yusuf Ismail Nakhoda, MT)
NIP.Y.1018800189


(Dr. Eng. Aryunto Soetedjo, ST, MT)
NIP. Y. 1030800417



Mengetahui,

Ketua Jurusan Teknik Elektro S-1


(Ir. Yusuf Ismail Nakhoda, MT)
NIP.Y. 1018800189

**KOSENTRASI TEKNIK ENERGI LISTRIK
JURUSAN TEKNIK ELEKTRO S -1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010**

PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI MULTI INKUBATOR MENGGUNAKAN PLC *SMART RELAY ZELIO LOGIC* BERBASIS SCADA

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Putra Hadi Prawira (0512011),
Ir. Yusuf Ismail Nakhoda, MT, Dr. Eng. Aryuanto Soetedjo, ST, MT
Institut Teknologi Nasional
Jl. Raya Karangploso KM.2 Malang
putra_php10@yahoo.co.id

Abstrak

Pada proses persalinan, sering terjadi seorang bayi lahir sebelum waktunya atau yang dinamakan dengan lahir prematur. Bayi yang lahir prematur perlu mendapatkan perawatan yang khusus dengan cara menyediakan inkubator bayi, dimana bayi prematur membutuhkan suhu antara 36°C - 37°C.

Kendala pada sistem inkubator yang ada dirumah sakit khususnya biasa yang berada didesa atau kota – kota yang belum maju, inkubator yang digunakan masih menggunakan inkubator yang dikontrol secara manual atau semi otomatis. Berangkat dari permasalahan tersebut maka diperukan sistem kontrol inkubator yang dapat dikontrol secara otomatis dan juga dapat dikontrol secara jarak jauh yaitu melalui SCADA.

Sistem kontrol inkubator zelio smart relay berbasis SCADA ini memiliki keistimewaan dapat mengontrol secara otomatis lebih dari satu inkubator dan juga dapat dikontrol dan dimonitor melalui Laptop/PC.

Kata Kunci : Bayi Prematur, *Smart relay*, SCADA (*Supervisory Control and Data Acquisition*).



KATA PENGANTAR

Dengan mengucap syukur kehadirat ALLAH SWT yang telah memberikan petunjuk, berkah dan rahmat-Nya, dan juga yang telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul :

**“ PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI
MULTI INKUBATOR MENGGUNAKAN PLC *SMART RELAY*
ZELIO LOGIC BERBASIS SCADA ”**

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata Satu (S-1) di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

- Kedua orang tua, Sulistiyo Hardi dan Sri Endah Ratna Sari, yang senantiasa memberikan doa dan dukungan dalam pengerjaan skripsi.
- Bapak Prof. Dr. Eng Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang.
- Bapak Ir. Sidik Noetjahjono, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
- Bapak Ir. F.Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S1 Konsentrasi Teknik Elektronika.
- Bapak Ir. Yusuf Ismail Nahkoda, MT dan Bapak Dr. Aryuanto Soetedjo, ST, MT selaku Dosen Pembimbing yang telah memberikan bimbingan,

pengarahan, serta ilmu-ilmu yang sangat berharga sehingga skripsi ini dapat terselesaikan.

- Teman – teman seangkatan yang telah banyak memberi kritik maupun saran dalam pengerjaan skripsi ini.
- Mas Irul sekeluarga yang memberi masukan dan tempat dalam pengerjaan skripsi ini.
- Someone yang senantiasa memberi kritikan setiap saat dan senantiasa menunggu hingga selesainya skripsi ini.
- Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohon maaf kepada semua pihak bila mana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, Agustus 2010

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	xii
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xii
 BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.1.2 Rumusan Masalah	1
1.1.3 Tujuan Penelitian.....	2
1.1.4 Batasan Masalah.....	2
1.1.5 Metodologi Pemecahan Masalah.....	3
1.6 Sistematika Penulisan	4
1.7 Relevansi.....	5
 BAB II LANDASAN TEORI	
2.1 Pendahuluan	6
2.2 PLC (<i>Programmable Logic Control</i>)	6

2.2.1	Struktur Dan Peralatan Perlengkapan PLC.....	7
2.2.2	Sistem Penulisan Program PLC	8
2.2.3	Konfigurasi Sistem PLC	10
2.2.3.1	<i>Central Procesing Unit (CPU)</i>	10
2.2.3.2	<i>Power Supply Unit</i>	13
2.2.3.3	<i>Input/Output Unit</i>	14
2.2.3.4	Data dan Memory PLC.....	16
2.2.4	Zelio Logic <i>Smart Relay</i>	20
2.2.4.1	Arsitektur Zelio Logic <i>Smart Relay</i>	24
2.2.4.2	Zelio Soft Software	25
2.3	Sensor Suhu.....	27
2.3.1	Sensor Suhu (IC LM 35).....	28
2.4	SCADA.....	29
2.4.1.	SCADA (Supervisory Control And Data Acquisition)	29
2.4.1.1	Definisi dan Sejarah SCADA	29
2.4.1.2	Komponen SCADA.....	32
2.4.1.2.1	Sistem Komunikasi SCADA.....	33
2.4.1.2.2	Software SCADA.....	34
2.4.1.3	Perkembangan SCADA.....	36
2.4.2.	WINLOG SCADA Lite.....	39
2.4.2.1	Tentang Winlog.....	39
2.4.2.2	Panduan Menggunakan Winlog	41
2.4.2.2.1.	Project Manager	41
2.4.2.2.2	Pengoperasian Project.....	41

2.4.2.2.2.1 Menjalankan project	41
2.4.2.2.2.2 Membuat project baru.....	42
2.4.2.2.2.3 Meng-copy project.....	42
2.4.2.2.2.4 Menghapus project	42
2.4.2.2.2.5 Mengekspor File.....	43
2.4.2.2.2.6 Mengimpor File.....	43
2.4.2.2.2.7 Membuka <i>windows</i> folder dari <i>project-project</i> yang ada	43
2.4.2.2.3 Konfigurasi.....	44
2.4.2.2.3.1 Options	44
2.4.2.2.3.2 Channel	45
2.4.2.2.3.3 Devices.....	45
2.4.2.2.4 Gate.....	46
2.4.2.2.5 Code.....	46
2.4.2.2.6 Template	47
2.4.2.3. <i>Gate Builder</i>	47
2.4.2.3.1 <i>Numerical gate</i>	48
2.4.2.3.1.1 <i>General</i>	48
2.4.2.3.1.2 <i>Sampling</i>	49
2.4.2.3.1.3 <i>Value</i>	50
2.4.2.3.1.4 <i>Conversion</i>	52
2.4.2.3.1.5 <i>Tolerance</i>	53
2.4.2.3.2 Digital gates	54
2.4.2.3.3 String gate	54
2.4.2.3.4 <i>Composed/Compound gate</i>	55

2.4.2.3.5 <i>String Gate</i>	56
2.4.2.3.6 Event/Alarm Gates	56
2.4.2.4. <i>Template Builder</i>	58
2.4.2.4.1 <i>Template window</i>	58
2.4.2.4.2 Deskripsi Komponen	60
2.4.2.4.2.1 BkBitmap	60
2.4.2.4.2.2 Bitmap.....	60
2.4.2.4.2.3 <i>Button</i>	62
2.4.2.4.2.4 Edit.....	63
2.4.2.4.2.5 <i>Gauge</i>	64
2.4.2.4.2.6 Label	65
2.4.2.4.2.7 Led.....	66
2.4.2.4.2.8 Metafile.....	66
2.4.2.4.2.9 <i>Switch</i>	66
2.5 Relay	67
2.6 MCB	70
2.7 Buzzer	72

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Pendahuluan	73
3.2 Perancangan Perangkat Keras (<i>Hardware</i>).....	73
3.2.1 Blok Diagram Sistem.....	74
3.2.2 Perancangan Inkubator	77
3.2.3 Perancangan Pemasangan Sensor Suhu (IC LM 35).....	77

3.2.4	Power Supply	78
3.3	Perancangan Perangkat Lunak (Software).....	79
3.3.1	Perancangan Software Zelio Soft 2	79
3.3.2	Perancangan Software Winlog Lite 2.06.74	83
3.3.3	Diagram Alir Perancangan	84
3.4	Spesifikasi Alat	85

BAB IV PENGUJIAN SISTEM

4.1	Pendahuluan	87
4.2	Pengujian Sensor Suhu	87
4.2.1	Tujuan	87
4.2.2	Peralatan yang Digunakan	87
4.2.3	Prosedur Pengujian	88
4.2.4	Hasil Pengujian	88
4.3	Pengujian <i>Power Supply</i>	90
4.3.1	Tujuan	90
4.3.2	Peralatan yang Digunakan	90
4.3.3	Prosedur Pengujian	90
4.3.4	Hasil Pengujian	91
4.4	Pengujian Daya Pada Sistem Kontrol.....	91
4.4.1	Tujuan	91
4.4.2	Peralatan yang Digunakan	92
4.4.3	Prosedur Pengujian	92
4.4.4	Hasil Pengujian	92

4.5	Pengujian Keseluruhan Sistem.....	93
4.5.1	Tujuan	93
4.5.2	Peralatan yang Digunakan	94
4.5.3	Prosedur Pengujian.....	94
4.5.4	Hasil Pengujian	95

BAB V PENUTUP

5.1	Kesimpulan	101
5.2	Saran	101

DAFTAR PUSTAKA	104
-----------------------------	------------

LAMPIRAN	105
-----------------------	------------

DAFTAR GAMBAR

Gambar 2.1 Diagram Blok PLC	7
Gambar 2.2 Struktur Dasar PLC.....	8
Gambar 2.3 Contoh Program Menggunakan <i>Ladder Diagram</i>	9
Gambar 2.4 Contoh Pemrograman Menggunakan FBD	10
Gambar 2.5 Sistem Komponen PLC.....	10
Gambar 2.6 Antarmuka Input.....	15
Gambar 2.7 Antarmuka Output	16
Gambar 2.8 <i>Type Compact Smart Relays</i>	21
Gambar 2.9 <i>Modular Smart Relay</i>	22
Gambar 2.10 <i>Discrete I/O Extension Modules</i>	23
Gambar 2.11 <i>Modbus communication module or Ethernet communication Module</i>	23
Gambar 2.12 Indikator kesalahan pemrograman pada Zelio Soft	26
Gambar 2.13 Kabel D-485/9D RS232 to RS485 Converter.	27
Gambar 2.14 Karakteristik Sensor Suhu	28
Gambar 2.15 Sensor Suhu IC LM 35.....	29
Gambar 2.16. Sistem “Sensor to panel” menggunakan arus 4-20 mA atau tegangan.	30
Gambar 2.17 Panel “Primitive” SCADA	30
Gambar 2.18 Komputer sebagai panel monitoring dan pengendalian pada SCADA	31
Gambar 2.19 Tipikal Sistem SCADA.....	32
Gambar 2.20 Sistem Komunikasi pada SCADA.....	33

Gambar 2.21 Stand alone PLC	36
Gambar 2.22 PLC dengan HMI.....	37
Gambar 2.23 Multiple PLC dengan HMI (RS-485 Multidrop).....	37
Gambar 2.24 Multiple PLC dengan HMI (Special Bus).....	37
Gambar 2.25 Multiple PLC dengan Multiple HMI (Special Bus dan TCP/IP).....	38
Gambar 2.26 Multiple PLC Multiple HMI (TCP/IP)	38
Gambar 2.27 Tampilan Project Manager	41
Gambar 2.28 Menjalankan project.....	42
Gambar 2.29 Mengekspor file	43
Gambar 2.30 Menu Configuration.....	44
Gambar 2.31 Konfigurasi Channel.	45
Gambar 2.32 Konfigurasi Device	46
Gambar 2.33 GateBuilder.....	47
Gambar 2.34 Parameter General pada Numerical gate.....	49
Gambar 2.35 Parameter Sampling pada Numerical gate.	50
Gambar 2.36 Parameter value pada Numerical gate.....	52
Gambar 2.37 Parameter <i>Conversion</i> pada <i>Numerical Gate</i>	52
Gambar 2.38 <i>Parameter Tolerance</i> pada <i>Numerical gate</i>	53
Gambar 2.39 <i>Parameter Operation</i> pada <i>Compound Gate</i>	56
Gambar 2.40 Parameter Condition pada Event Gate.	57
Gambar 2.41 Template window.	59
Gambar 2.42 Bentuk Fisik Relay.....	68
Gambar 2.43 konstruksi relay jenis kontak tukar	69
Gambar 2.44 jenis – jenis relay	70

Gambar 2.45 Bentuk Fisik MCB (<i>Magnetic Circuit Breaker</i>	70
Gambar 2.46 Bentuk Fisik Buzzer.....	72
Gambar 3.1 Blok Diagram Sistem Pengendalian Multi Inkubator Menggunakan LC <i>Smart Relay Zelio Logic</i> Berbasis SCADA.....	74
Gambar 3.2 Bentuk Fisik Inkubator.....	77
Gambar 3.3 Perancangan letak pemasangan sensor suhu IC LM 35 pada setiap Inkubator	77
Gambar 3.4 <i>Power Supply</i> Tegangan Untuk Sistem.....	79
Gambar 3.5 Konfigurasi Smart Relay	80
Gambar 3.6 Konfigurasi Modul Tambahan	81
Gambar 3.7 Konfigurasi Ladder dan Function Block Diagram (FBD)	82
Gambar 3.8 Software Winlog Lite 2.06.74	83
Gambar 3.9 Flowchart Sistem Inkubator 1 dan Inkubator 2.....	85
Gambar 4.1 Pengujian Rangkaian Sensor Suhu IC LM 35.....	88
Gambar 4.2 Rangkaian Pada IC LM35	88
Gambar 4.3 Hasil Pengujian Sensor IC LM 35	89
Gambar 4.4 Rangkaian Pengujian <i>Power Supply</i> 24 VDC dan 14 VDC.....	91
Gambar 4.5 Hasil Pengujian Power Supply 24 VDC dan 14 VDC.....	91
Gambar 4.6 Pengukuran Tegangan Suplai PLN.....	92
Gambar 4.7 Pengukuran Arus Yang Mengalir Pada Sistem	93
Gambar 4.8 Gambar Proses <i>Smart Relay Zelio Logic SR3</i> Aktif/ <i>Standby</i>	95
Gambar 4.9 Gambar Proses <i>Smart Relay Zelio Logic SR 3</i> Stop.	96
Gambar 4.10 Gambar Proses <i>Smart Relay Zelio Logic SR3</i> <i>Running/Bekerja</i> .96	
Gambar 4.11 Tampak samping Perancangan dan Pembuatan Sistem	

Pengendalia Multi Inkubator menggunakan PLC Smart Relay Zelio
Logic Berbasis SCADA 100
Gambar 4.12 Gambar kontrol SCADA (software winlog) 101

DAFTAR TABEL

Tabel 4.1 Hasil Pengujian Sensor Suhu IC LM35	89
Tabel 4.2 Hasil Pengujian Keseluruhan Sistem.....	97

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada suatu proses bersalin, sering terjadi seorang bayi lahir sebelum waktunya atau yang dinamakan dengan lahir premature. Bayi yang lahir prematur perlu mendapatkan perawatan dengan cara menyediakan inkubator bayi. Saat ini inkubator bayi yang ada di pasaran umumnya hanya menggunakan kontrol secara otomatisasi sensor suhu saja. Tentu hal ini mengakibatkan perawat harus setiap waktu mengawasi inkubator tersebut. Di dalam skripsi ini dicoba dirancang sebuah inkubator bayi yang dapat dikontrol secara otomatis serta juga dapat dikontrol secara jarak jauh agar perawat bisa mengawasi dan mengontrol dari ruangnya tanpa perlu mendatangi inkubator tersebut.

Inkubator akan dibuat dengan cara sederhana, pengaturan suhu dilakukan dengan memasang dua buah fan dan heater listrik dengan sistem kontrol PLC *Smart Relay Zelio Logic* berbasis SCADA agar dapat tetap menjaga suhu dan dapat memonitor apakah inkubator tersebut sudah bekerja dengan seperti yang diharapkan atau tidak melalui SCADA (software Winlog). Inkubator bayi ini diatur suhu tubuh bayi dijaga agar berada pada suhu 36°C - 37°C agar suhu bayi sesuai dengan suhu dalam rahim ibu. [5]

1.1.2 Rumusan Masalah

Dalam perancangan dan pembuatan sistem pengendalian suhu inkubator menggunakan PLC *Smart Relay Zelio Logic* berbasis SCADA, dapat dirumuskan beberapa masalah yang akan dibahas yaitu :

1. Bagaimana merancang dan membuat perangkat keras atau hardware dari sistem pengendali suhu menggunakan PLC *Smart Relay Zelio Logic* berbasis SCADA.
2. Bagaimana merancang perencanaan letak pemasangan sensor suhu berdasarkan fungsinya pada inkubator.
3. Bagaimana men-*setting* agar deteksi sensor suhu tidak kurang atau melampaui suhu yang diperlukan.
4. Bagaimana merancang perangkat lunak atau *software* dari otomatisasi sensor suhu menggunakan PLC *Smart Relay Zelio Logic* berbasis SCADA.

1.1.3 Tujuan Penelitian

Berdasarkan permasalahan yang dikemukakan di atas maka, tujuan dalam penulisan skripsi ini adalah untuk mengatur suhu inkubator secara otomatisasi dan jarak jauh agar dapat memudahkan dan memaksimalkan tugas perawat.

1.1.4 Batasan Masalah

Agar Perancangan dan Pembuatan Sistem Pengendalian Multi Inkubator Menggunakan *Programmable Logic Controler Smart Relay Zelio* berbasis

SCADA ini tidak meluas, maka penulis perlu untuk memberikan batasan – batasan terhadap masalah yang akan dibahas yaitu :

1. Tipe *Smart Relay* yang digunakan adalah *Zelio Logic SR3*.
2. Tipe *software* yang digunakan adalah *Zelio Soft 2*.
3. Tidak membahas sensor suhu secara detail.
4. Tidak membahas inkubator secara detail.
5. Tidak membahas jenis – jenis bayi premature.
6. Hanya membahas mengenai sistem otomatisasi dan pengontrolan Inkubator menggunakan *Programmable Logic Controler Smart Relay Zelio* berbasis SCADA.

1.1.5 Metodologi Pemecahan Masalah

Metode yang digunakan dalam pembahasan skripsi ini adalah:

1. Studi literature

Mencari referensi – referensi yang berhubungan dengan perencanaan dan pembuatan alat yang akan dibuat.

2. Perancangan Sistem

Sebelum melaksanakan pembuatan terhadap sistem, dilakukan perancangan terhadap sistem yang meliputi merancang rangkaian untuk tiap – tiap inkubator dan rangkaian keseluruhan sistem, serta perancangan terhadap *software*.

3. Pembuatan Sistem

Pada tahap ini realisasi alat yang dibuat, dilakukan perakitan sistem terhadap seluruh hasil rancangan yang telah dibuat.

4. Pengujian Sistem

Untuk mengetahui cara kerja alat, maka dilakukan pengujian tiap inkubator dan pengujian secara keseluruhan.

5. Pengolahan Data

Mengolah data dan menganalisa hasil pengujian sistem untuk membuat kesimpulan.

1.6 Sistematika Penulisan

Untuk mendapatkan arah yang tepat mengenai hal-hal yang akan dibahas maka dalam skripsi ini disusun sebagai berikut :

BAB I : PENDAHULUAN

Dalam Bab ini berisikan Latar Belakang, Rumusan Masalah, Tujuan, Batasan Masalah, Metodologi Penelitian, dan Sistematika Penulisan yang digunakan dalam pembuatan tugas akhir ini.

BAB II : LANDASAN TEORI

Pada Bab ini dibahas tentang teori – teori yang mendukung dalam perencanaan dan pembuatan alat ini.

BAB III : PERENCANAAN SISTEM

Dalam Bab ini akan dibahas mengenai perencanaan dan pembuatan tugas akhir ini yang meliputi seluruh sistem ini baik perangkat keras maupun perangkat lunak sistem.

BAB IV : PENGUJIAN SISTEM

Dalam Bab ini membahas tentang pengujian dan hasil yang diperoleh dari sistem yang telah dibuat.

BAB V : KESIMPULAN DAN SARAN

Dalam Bab ini berisi kesimpulan – kesimpulan yang diperoleh dari perencanaan dan pembuatan tugas akhir ini serta saran – saran guna menyempurnakan dan mengembangkan sistem lebih lanjut.

1.7 Relevansi

Manfaat yang didapatkan dalam skripsi ini adalah :

- Membantu rumah sakit kecil, rumah bersalin atau bidan khususnya yang berada dipelosok desa agar dapat menggunakan inkubator tanpa harus dihantui dengan perasaan bahwa inkubator itu mahal
- Bagi masyarakat sebagai pasien, dapat mengurangi biaya rumah sakit bila bayi mereka lahir secara premature

Sedangkan dampak skripsi ini baik yang positif maupun negatif adalah :

- Mengurangi jumlah angka kematian bayi yang lahir premature
- Pengeluaran biaya rumah sakit, rumah bersalin untuk pengadaan inkubator berkurang karena tidak perlu lagi mengeluarkan biaya yang mahal
- Bagi perusahaan pembuat inkubator akan merugi tetapi dampak positifnya mereka akan berlomba – lomba membuat inkubator yang bagus tapi dengan harga yang murah

BAB II

LANDASAN TEORI

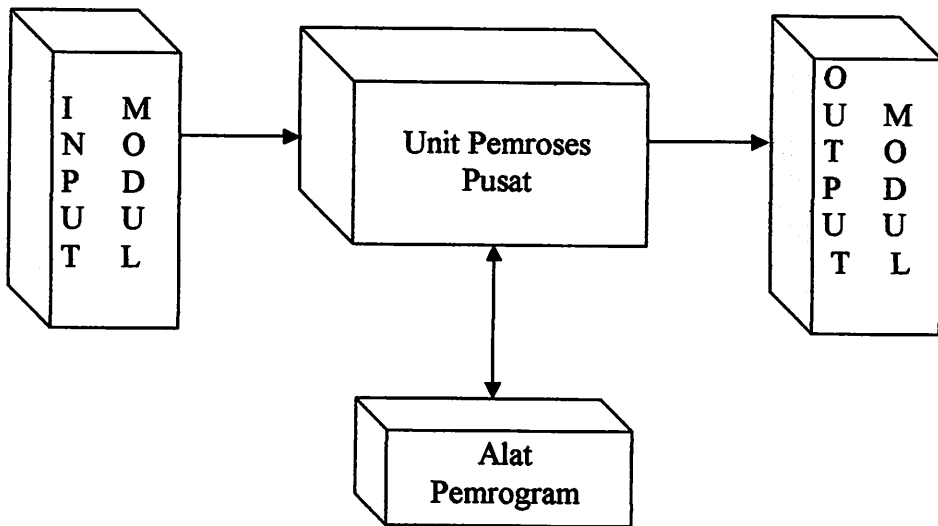
2.1 Pendahuluan

Pada bab ini akan dibahas mengenai teori penunjang dari peralatan yang direncanakan. Teori penunjang ini akan membahas tentang komponen dan peralatan pendukung pada alat yang dibuat. Pokok pembahasan pada bab ini adalah :

1. PLC (*Programmable Logic Control*)
2. Sensor Suhu
3. SCADA
4. Relay
5. MCB
6. Buzzer

2.2 PLC (*Programmable Logic Control*) ^[1]

PLC pada dasarnya adalah sebuah computer yang khusus dirancang untuk mengontrol suatu proses atau mesin. Proses yang dikontrol ini dapat berupa regulasi variable secara kontinyu seperti pada sistem – sistem servo, atau hanya melibatkan kontrol 2 keadaan (On/Off) saja, tetapi dilakukan secara berulang – ulang seperti umum dijumpai pada mesin pengeboran, sistem konveyor dan lain sebagainya.



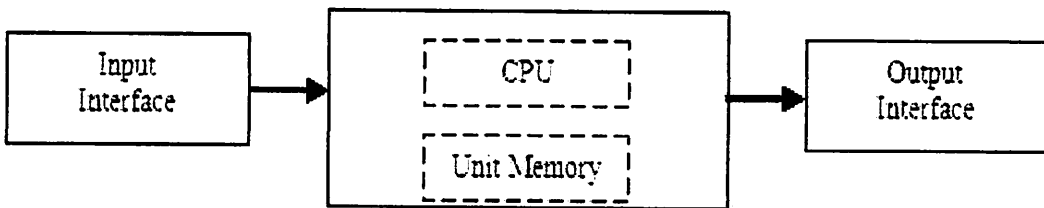
Gambar 2.1
Diagram Blok PLC ^[1]

2.2.1 Struktur Dan Peralatan Perlengkapan PLC ^[1]

Pada umumnya PLC dapat dibayangkan sebagai sebuah personal computer konvensional karena konfigurasi internal yang ada pada PLC mirip dengan konfigurasi yang dimiliki oleh sebuah personal komputer. Secara khusus, PLC dirancang untuk menangani suatu sistem kontrol otomatis pada mesin – mesin industri atau aplikasi – aplikasi lain di industri seperti kontrol lampu lalu lintas, air mancur, sistem bagasi lapangan terbang, *Water Level Controlling (WLC)* dan lain – lain.

Secara garis besar struktur dasar PLC dapat dibagi menjadi empat kelompok komponen utama yang terdiri dari antar muka (*interface*) input, antar muka (*interface*) output, Unit Pemroses (*Central Processing Unit/CPU*) dan Unit Memory. Dalam CPU sebuah PLC dapat diibaratkan sebuah kumpulan ribuan *relay* walaupun kenyataannya bukan berarti terdapat ribuan *relay* berskala kecil. Tetapi dalam PLC berisi rangkaian elektronika digital yang berfungsi sebagai

Contact Normally Open (NO) dan *Normally Close (NC) relay*. Satu kontak NO dan NC pada PLC dapat digunakan berkali – kali untuk semua jenis instruksi dasar PLC kecuali *Output*. Instruksi *output* PLC tidak dapat dilakukan untuk nomor kontak yang sama.



Gambar 2.2
Struktur Dasar PLC [1]

Peralatan *input (Input Devics)* yang banyak digunakan sebagai sinyal *interface* sebuah PLC dapat berupa saklar – saklar atau sensor – sensor. Diantara sekian banyak peralatan *input* yang dipakai diantaranya *Pust Button, Limit Switch, Tumb Wheel Switch, Level Switch, Flow Switch* dan saklar tekan lainnya.

Yang termasuk peralatan kontrol (*Control Devices*) terdapat didalam PLC itu sendiri dan dapat deprogram ulang sesuai dengan sistem kontrol yang kita inginkan, peralatan *Controller* yang dimiliki oleh sebuah PLC dapat berupa *Internal Relay (Relay Coil), Latchng, Timer Coil, Counter, Electronic Card* dan lain – lain.

2.2.2 Sistem Penulisan Program PLC [1]

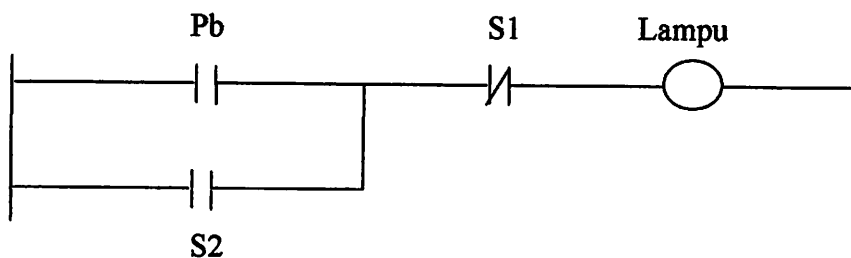
Pemrograman adalah penulisan serangkaian perintah yang memberikan instruksi pada PLC untuk melaksanakan tugas yang telah ditentukan. Sistem pemrograman sebuah PLC terdiri dari beberapa format yaitu :

1. *Ladder Diagram*

Penulisan dengan cara *Ladder Diagram* ini paling banyak digunakan pada sistem kontrol yang menggunakan *relay – relay* atau pada sistem yang menggunakan PLC, sehingga PLC penulisan *Ladder Diagram* ini merupakan pengembangan dari penulisan dan penggambaran rangkaian dalam sistem kontrol *relay* elektronik.

Penulisan dengan *Ladder Diagram* bertujuan untuk menampilkan urutan – urutan kerja sinyal listrik. Melalui diagram dapat diperlihatkan hubungan antara peralatan aktif atau tidak aktif (hidup atau mati) sesuai dengan urutan yang ditentukan.

Contoh penulisan program menggunakan *ladder* diagram seperti gambar :



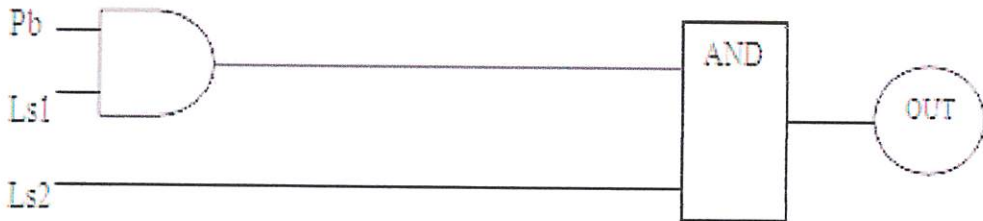
Gambar 2.3
Contoh Program Menggunakan *Ladder Diagram* ^[1]

2. *Function Block Diagram (FBD)*

Penulisan program menggunakan FBD memiliki persamaan dengan *Ladder Diagram*, yaitu kedua cara ini sama – sama digambarkan dalam bentuk grafik. Penggambaran atau penulisan program dengan cara ini biasanya dilakukan untuk sistem program scanning dan untuk

menggambarkan sistem program sekuensial. Cara ini juga dapat digunakan sebagai *Flow Chart*.

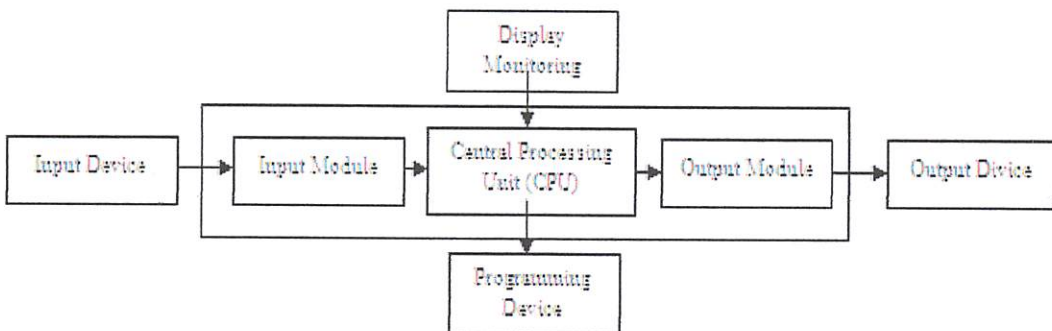
Simbol yang dapat digunakan dalam sistem FBD berupa symbol – simbol gerbang logika seperti gambar :



Gambar 2.4
Contoh Pemrograman Menggunakan FBD ^[1]

2.2.3 Konfigurasi Sistem PLC ^[1]

Komponen – komponen PLC yang diperlukan untuk sistem kendali diantaranya berupa *Central Controller Unit (CCU)* yang disebut dengan istilah *Central Processing Unit (CPU)* yang terdiri dari Prosesor, Memori, dan *Power Supply* serta bagian *Input/Output (I/O) Structure* dan *Program Device*.



Gambar 2.5
Sistem Komponen PLC ^[1]

2.2.3.1 *Central Procesing Unit (CPU)* ^[1]

Unit pemroses utama atau lebih dikenal dengan *Central Procesing Unit* berfungsi untuk mengambil instruksi dari memori, pengkodean kemudian

mengeksekusi instruksi tersebut. Selama proses tersebut CPU akan menentukan putusan untuk pengendalian atau menghasilkan fungsi aritmatika dan logika serta mendeteksi sinyal dari luar CPU.

Pada dasarnya unit pemroses utama (CPU) terdiri dari :

a. Register

Register merupakan penyimpanan data sementara yang dapat digunakan selama pengeksekusian program. Register ini akan mempercepat suatu proses, karena data yang sering dipakai diletakkan pada register sehingga bila CPU memerlukannya tidak perlu membaca dari memori.

b. Control Unit (CU)

Unit pengendali atau *Control Unit* (CU) mengendalikan atau mengarahkan urutan operasi pada prosesor dan mengirim sinyal pengendali untuk mengkoordinasikan aliran informasi dan data antara bagian pada prosesor seperti mentransfer atau sebaliknya, mengambil data dari input image table, mengirim data ke output image table dan operasi – operasi lain dalam prosesor. Disamping itu unit kendali juga member respon terhadap sinyal dari luar.

c. Aritmatic Logical Unit (ALU)

Unit logika dan aritmatika atau Aritmatika Logical Unit (ALU) berfungsi untuk melakukan operasi – operasi logika dan aritmatika seperti penjumlahan, perkalian, pembagian dan logika dalam satu program.

Biasanya PLC menggunakan *Chip Microprosesor* sebagai intinya dan sekaligus merupakan otak dari PC. Gerakan *Actuator* yang diperintahkan oleh inti ini dalam bentuk program yang diolah oleh *Microprocessor*. Jenis *Microprocessor*

umum digunakan adalah : Z80, 6800, 8086, 6502, 68000, 80286, 80386, atau pun 80486 serta yang lainnya sampai generasi sekarang *Intel Pentium*.

Karakteristik terpenting dari PLC adalah kemudahan pemakaian dalam menggantikan program dengan mudah dan cepat. Tujuan ini dapat dicapai dengan membuat karakteristik PLC dilengkapi dengan sistem memori. Sistem memori ini dimaksudkan untuk menyimpan data – data urutan instruksi ataupun program yang dapat dieksekusi oleh prosesor sesuai dengan perintah yang telah diberikan dalam program.

Sistem memori PLC terdiri dari dua virtual memori, meliputi :

1. *Executive Memory*

Memory ini tersusun dari sekumpulan program – program permanen yang dianggap sebagai bagian dari PLC. Program permanen ini mengarahkan atau menjalankan aktifitas seluruh sistem seperti eksekusi program, komunikasi peralatan dan lain – lain. Dengan kata lain *executive memory* adalah *memory* yang dapat menyimpan instruksi – instruksi *software*, seperti instruksi *relay*, *block transfer* instruksi matematik dan lain – lain. Daerah *memory* ini tidak dapat diakses oleh *user* atau pemakai.

2. *Application Memory*

Siste ini berguna untuk menyimpan dan tempat menampung instruksi – instruksi program yang diinput oleh pemakai. *Memory* ini terdiri dari beberapa bagian yang memiliki fungsi dan penggunaan yang khusus.

2.2.3.2 *Power Supply Unit* ^[1]

Unit PLC tidak akan bekerja jika tidak diberi energi. Energi yang digunakan untuk menghidupkan PLC dapat berupa sumber AC 120 Volt atau 240 Volt dan dapat juga ditentukan sumber arus DC 5 Volt sampai dengan 10 Volt. Untuk menghidupkan PLC, pemakai tinggal menyambungkan bagian input energi dengan tegangan dan arus listrik yang sesuai.

Selain menyediakan tegangan listrik, *Power Supply* juga dapat memonitor dan memberikan sinyal kepada CCU apabila terjadi suatu kesalahan. Dengan kata lain, *Power Supply* selain sebagai pemberi daya, berfungsi juga sebagai proteksi komponen sistem. Perlu diperhatikan bahwa komponen *Power Supply* jangan dihubungkan dengan sumber yang tidak stabil.

Power Supply yang baik dan idealnya dirancang untuk mengamankan terjadinya fluktuasi kondisi daya. Tetapi sebuah *Power Supply* belum tentu dapat mengkompensasi kondisi ketidak stabilan tegangan yang terjadi.

Ketidak stabilan tegangan ini, biasanya disebabkan oleh :

- Jauhnya lokasi sumber energi
- Sistem sambungan yang tidak baik
- Dekat dengan peralatan berat

Untuk mengatasi hal tersebut, diperlukan adanya suatu alat yang dapat menstabilkan tegangan sebelum digunakan. Alat yang biasa dipakai adalah *Constant Voltage Transformer* atau lebih dikenal dengan nama *Stabilizer*.

Untuk mengatasi masalah lain yang akan mempengaruhi jalannya program PLC, maka sebaiknya PLC dilengkapi atau dijauhkan dengan peralatan lain yang dapat menimbulkan efek elektromagnetik.

2.2.3.3 *Input/Output Unit* ^[1]

Input/Output Unit adalah struktur masukan keluaran yang terdapat dalam PLC dan menyebabkan PLC tersebut dapat bekerja atau menjalankan instruksi programnya. Sebagaimana fungsinya PLC sebagai pengontrol suatu proses operasi mesin, maka struktur *Input/Output* merupakan perantara atau bagian yang menghubungkan antara bagian kontrol seperti saklar motor *starte*, katup – katup dan sebagainya dengan CCU-nya.

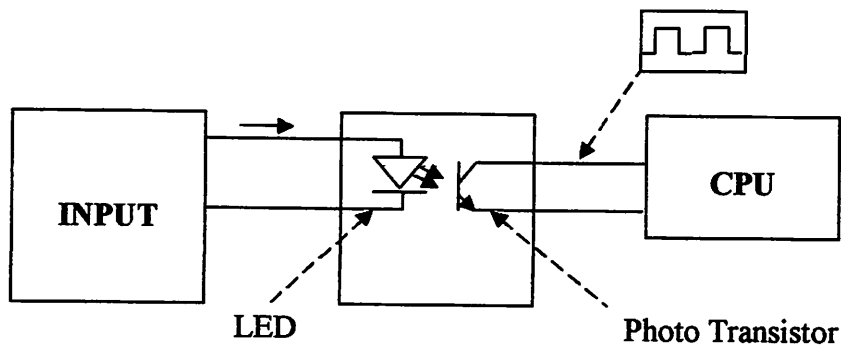
Sinyal yang diolah oleh CCU merupakan sinyal elektrik bertegangan rendahberkisaran 5 Volt DC. Sedangkan sinyal dari elemen kontrol dapat berupa arus DC atau arus AC yang bertegangan sapaai dengan 240 Volt.

Sinyal *Input/Output* yang bias diterima oleh PLC terdiri dari dua macam sinyal, yaitu sinyal analog dan sinyal digital (*binary*). Sinyal *binary* adalah sinyal yang mempunyai kondisi hidup “1” dan mati “0”. Sinyal ini merupakan sinyal ang paling umum digunakan dalam sistem PLC. Sedangkan sinyal analog adalah sinyal yang dapat berubah – ubah setiap saat, misalnya berupa arus atau keadaan tegangan listrik pada saat – saat tertentu. Untuk kebanyakan PLC saat ini, penggunaan sinyal analog harus diterjemahkan terlebih dahulu menjadi sinyal binary menggunakan suatu modul input analog.

1. *Input Analog*

Sebelum peralatan modul input dihubungkan dengan modul input analog, terlebih dahulu peralatan ini harus dihubungkan dengan sebuah tranduser, atau transmitter yang berfungsi sebagai pengubah sinyal analog yang berasal dari peralatan input kedalam bentuk sinyal analog arus DC.

Terjadinya perubahan sinyal ini kan membandingkan antara variable yang diukur dengan tegangan yang diterima oleh modul akan diubah ke dalam bentuk sinyal *binary*. Perubahan bentuk sinyal ini dilakukan oleh suatu alat yang dinamakan *Analog to Digital Converter (A/D atau ADC)*.



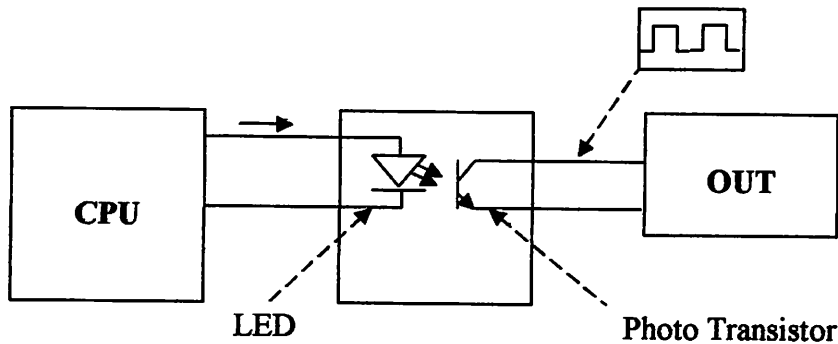
Gambar 2.6
Antarmuka Input ^[1]

Besarnya sinyal tegangan yang masuk dibagi ke dalam beberapa bagian yang dinamakan *Count* atau hitungan system digital. Perubahan hitungan digital terkecil yang mampu dibentuk dengan istilah resolusi. Contoh resolusi ADC 12 bit, ini berarti sinyal inputannya dapat dibagi kedalam 12 bit. Nilai didapatkan berdasarkan perhitungan 2^{12} atau dengan jumlah bilangan decimal yang berkisaran 0 sampai dengan angka 4095.

2. *Output Analog*

Modul ini digunakan untuk mengontrol peralatan yang menerima sinyal tegangan / arus kontinu (*analog*). Sebagaimana input analog, modul output analog dihubungkan dengan alat pengontrol melalui transduser atau *transmitter*. *Transmitter* ini berfungsi untuk memperbesar atau memperkecil serta mengubah sinyal *output* yang berupa *binary* ke dalam bentuk sinyal yang dapat diterima oleh peralatan output.

Perbedaan antara *modul input* dan modul output terletak pada jenis konverter yang digunakan. Kalau *modul input* menggunakan A/D converter, sedangkan *modul output* menggunakan D/A konverter atau DAC (*Digital to Analog Converter*). Besaran analog output yang dihasilkan akan sebanding dengan besaran \square umeric yang diterimanya. Jadi selain menghitung atau menangkap sinyal, DAC akan menghasilkan sinyal analog yang besarnya sebanding dengan arus / tegangan minimum dan maksimum.



Gambar 2.7
Antarmuka Output ^[1]

2.2.3.4 Data dan Memory PLC ^[1]

Memori PLC terdiri dari :

1. IR (*Internal Relay*)

Internal Relay mempunyai pembagian fungsi seperti IR input, IR output dan IR work area untuk pengelolah data pada progam IR input dan output adalah IR yang berhubungan dengan terminal input dan output pada PLC. Sedangkan IR work area tidak dihubungkan ke terminal PLC, tetapi terletak pada *internal memory* PLC dan berfungsi untuk pengelolah logika program (manipulasi program).

Selain itu terdapat juga IR yang difungsikan untuk SYSMAC BUS Area, Spesial I/O unit area, Optical I/O unit area dan *Group 2 high density* I/O unit area, dengan fungsi masing – masing sebagai berikut :

- SYSMAC BUS Area berfungsi untuk komunikasi data PLC antara CPU PLC dan I/O unit PLC dengan hanya menggunakan dua kabel saja yaitu RS 485 dengan jarak maksimum 200 meter.
- Special I/O area berupa IR yang digunakan oleh spesial I/O unit PLC, contoh : *analog input*, *analog output*, dan lain – lain yang berfungsi untuk menyimpan dan mengolah datanya.
- Optical I/O area berupa IR yang digunakan untuk mengolah data dan menyimpan data dari optical I/O unit PLC.
- *Group 2 high density* I/O unit area berupa IR yang berfungsi untuk menyimpan dan mengolah data dari *high density* I/O unit group 2.

2. SR (*Special Relay*)

Relay yang mempunyai fungsi khusus seperti *flags*, misalnya pada instruksi penjumlahan, terdapat kelebihan digit pada hasilnya (*Carry Flags*), kontrol bit PLC, informasi kondisi PLC dan sistem *clock* (pulsa 1 detik, 0,2 detik dan lain – lain).

3. AR (*Auxiliary Relay*)

Terdiri atas *flags* dan bit dengan tujuan – tujuan khusus dan dapat menunjukkan PLC yang disebabkan oleh kegagalan sumber tegangan, kondisi spesial I/O, kondisi I/O unit, kondisi CPU PLC, kondisi memori PLC dan lain – lain.

4. HR (*Holding Relay*)

Berfungsi untuk menyimpan data (bit – bit penting) karena tidak akan hilang walaupun sumber tegangan PLC telah terputus (*off*).

5. LR (*Link Relay*)

Digunakan untuk data link pada PLC link sistem. Artinya berfungsi untuk tukar menukar informasi antara dua PLC atau lebih dalam suatu sistem kontrol yang saling berhubungan satu dengan yang lainnya dan menggunakan banyak PLC (minimum 2 PLC).

6. TR (*Temporary Relay*)

Berfungsi untuk menyimpan sementara kondisi logika program yang terdapat pada ladder diagram yang mempunyai titik percabangan khusus.

7. TC (*Timer / Counter*)

Timer digunakan untuk mendefinisikan sistem waktu tunda (*time delay*), sedangkan Counter digunakan sebagai penghitung. Timer dalam suatu PLC mempunyai orde 100 ms dan ada juga yang mempunyai orde 10 ms seperti TIMH (15). Untuk TIM 000 sampai dengan 015 dapat dioperasikan secara *interrupt* untuk mendapatkan waktu yang lebih presisi.

8. DM (*Data Memory*)

Data memory berfungsi untuk menyimpan data – data program, karena isi DM tidak akan hilang (*reset*) walaupun sumber tegangan PLC off.

Ada beberapa macam DM, diantaranya :

- *DM Read / Write* : DM ini dapat dihapus dan ditulis oleh program yang kita buat, jadi sangat berguna untuk memanipulasi data program.
- *DM Spesial I/O Unit* : DM ini berfungsi untuk menyimpan dan mengelolah hasil spesial I/O unit, mengatur dan mendefinisikn sistem kerja spesial I/O unit.
- *DM History Log* : DM ini dapat menyimpan informasi – informasi penting pada saat PLC terjadi kegagalan sistem operasinya. Pesan – pesan kesalahan yang terjadi dalam sistem PLC dapat disimpan berupa kode – kode angka tertentu.
- *DM Link Test Area* : DM ini berfungsi untuk menyimpan informasi – informasi yang menunjukkan status dari sistem link PLC.
- *DM setup* : Berfungsi untuk setup kondisi default (kondisi kerja saat PLC aktif). Pada DM inilah kemampuan kerja dari suatu PLC didefinisikan terlebih dahulu sebelum PLC tersebut deprogram dan dioperasikan pada suatu sistem kontrol dan setupnya disesuaikan dengan sistem kontrol yang diinginkan.

9. UM (*Upper Memory*)

Memori ini berfungsi untuk menyimpan dan menjalankan program kita (*user program*) yang mempunyai kapasitas tergantung pada masing – masing tipe PLC yang dipakai.

Semua *memory* (selain DM dan UM) dapat berfungsi sebagai sebuah relay yang mempunyai coil, kontak NO dan kontak NC. Begitu juga timer dan counter yang dapat berfungsi sama seperti timer dan counter pada umumnya yang

mempunyai kontak NO dan kontak NC. Sedangkan DM tidak mempunyai bentuk, tetapi hanya berupa *channel / word* saja. DM dapat difungsikan untuk menyimpan data – data penting yang tidak boleh hilang, pada saat power telah off atau berfungsi untuk memanipulasi program yang kita buat. Selain itu memori yang mempunyai sifat dapat menyimpan data program jika listrik mati adalah DM dan HR sedangkan yang lain kembali *reset* (hilang / terhapus).

2.2.4 Zelio Logic *Smart Relay* ^[3]

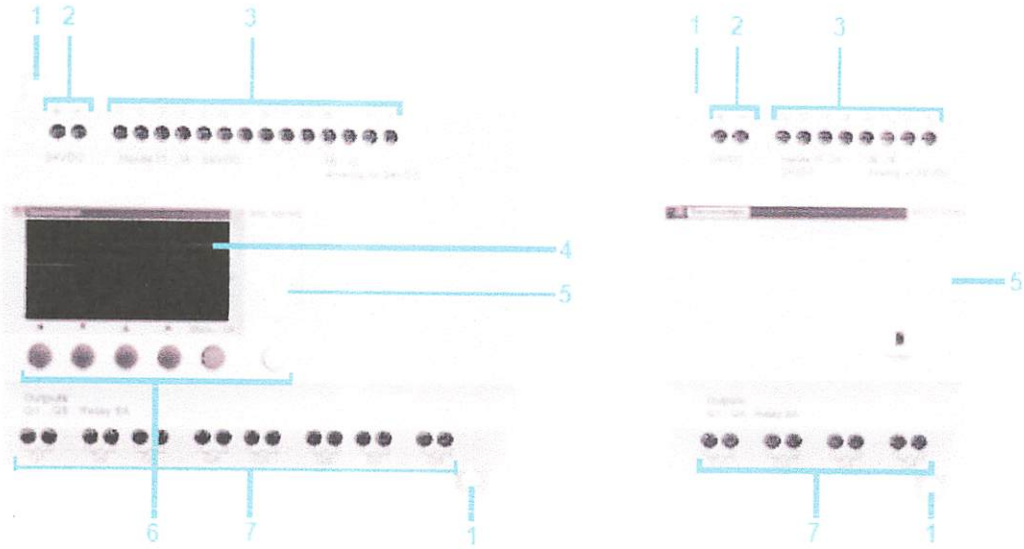
Zelio Logic *Smart Relay* didesain untuk automasi terhadap sistem yang tidak terlalu kompleks. Zelio Logic *Smart Relay* bias digunakan untuk aplikasi dibidang industri dan komersial.

Dalam sektor industri, Zelio Logic *Smart Relay* biasanya digunakan untuk automasi dibidang *Finishing Production*, automasi mesin pengepakan dan perakitan, pelastik dan material *Processing sector* dan automasi sistem untuk mesin – mesin yang bersifat pekebunan atau *Agricultural* (seperti irigasi pengairan, mesin pompa dan *Green House*).

Dalam sektor komersial atau bangunan, Zelio Logic *Smart Relay* biasanya digunakan otomasi terhadap sistem parker gedung, pengontrolan lift maupun escalator, otomasi sistem keamanan serta otomasi terhadap sistem *compressor* dan *air conditioning*.

Zelio Logic *Smart Relay* memiliki 2 tipe yaitu tipe *Compact Smart Relay* dan tipe *Modular Smart Relay*. Untuk tipe *Compact Smart Relay* biasa digunakan untuk sistem automasi yang tidak terlalu kompleks dimana *input/output* yang dimiliki tipe *Compact Smart Relay* berjumlah hingga 20 I/O. apabila dibutuhkan

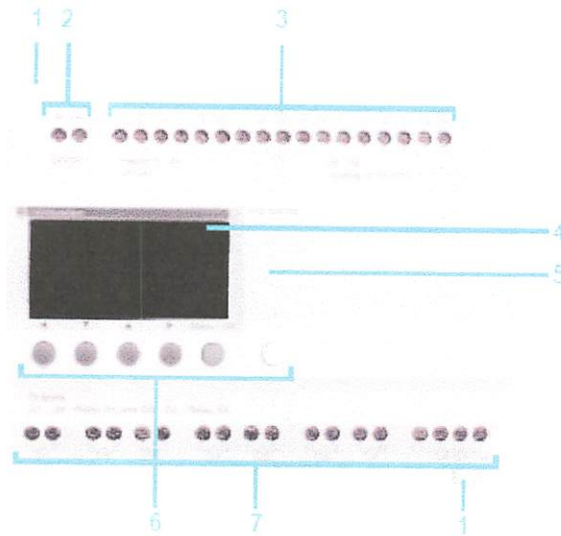
sistem yang lebih kompleks maka bias menggunakan tipe *Modular Smart Relay* yang dapat dipasangkan dengan *I/O Modul Extension* yang mencapai 6, 10 sampai 40 I/O dan sebuah modul komunikasi sesuai dengan fleksibilitas terhadap sistem yang dicapai.



Gambar 2.8
Type Compact Smart Relays ^[4]

Zelio Logic compact smart relays have the following on their front panel :

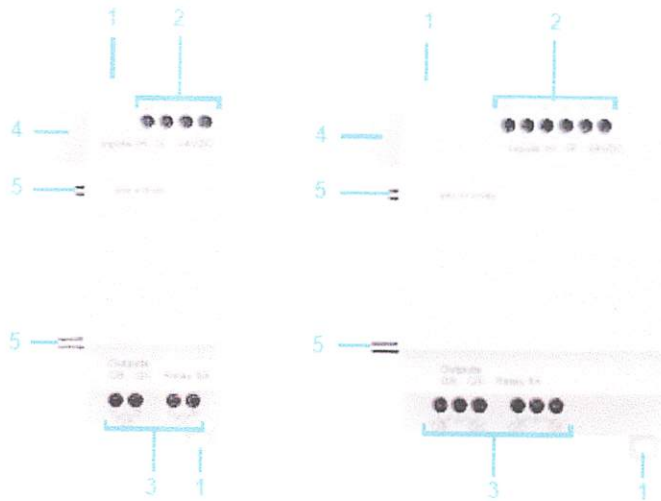
1. *Two retractable fixing lugs.*
2. *Two power supply terminals.*
3. *Terminals for connection of the inputs.*
4. *Backlit LCD display with 4 lines of 18 characters.*
5. *Slot for memory cartridge or connection to a PC or Modem communication interface.*
6. *6 buttons for programming and parameter entry.*
7. *Terminals for connection of the outputs.*



Gambar 2.9
Modular Smart Relay^[4]

Zelio Logic compact smart relays have the following on their front panel :

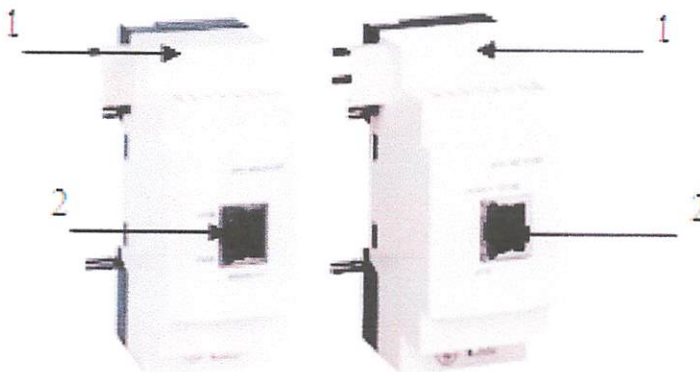
- 1. Two retractable fixing lugs.*
- 2. Two power supply terminals.*
- 3. Terminals for connection of the inputs.*
- 4. Backlit LCD display with 4 lines of 18 characters.*
- 5. Slot for memory cartridge or connection to a PC or Modem communication interface.*
- 6. 6 buttons for programming and parameter entry.*
- 7. Terminals for connection of the outputs.*



Gambar 2.10
Discrete I/O Extension Modules ^[4]

Discrete I/O extension modules have the following on their front panel :

1. Two retractable fixing lugs.
2. Terminals for connection of the inputs.
3. Terminals for connection of the outputs.
4. A connector for connection to the Zelio Logic smart relay (powered via the Zelio Logic smart relay).
5. Locating pegs.



Gambar 2.11
Modbus communication module or Ethernet communication module ^[4]

1. Modbus slave or Ethernet servercommunication module.
2. RJ45 connector for Modbus or Ethernet network connection.

2.2.4.1 Arsitektur Zelio Logic *Smart Relay* ^[1]

Arsitektur dari Zelio Logic *Smart Relay* antara lain :

1. Untuk *Power Supply*, Zelio Logic *Smart Relay* membutuhkan *Supply* tegangan sebesar 12 VDC, 24 VDC, 24 VAC dan 100 – 240 VAC.
2. Pemrograman dapat langsung menggunakan tombol pada *Smart Relay* dengan menggunakan bahasa *ladder* dan juga dapat melalui PC (*Personal Computer*). Apabila pemrograman dilakukan melalui PC, maka program dapat menggunakan bahasa ladder ataupun *Function Block Diagram*.
3. Untuk Zelio Logic *Smart Relay* SR3 memiliki terminal *input / output* sebanyak 10 I/O dan 26 I/O, sedangkan untuk Zelio Logic *Smart Relay* SR2 memiliki terminal *input / output* sebanyak 10 I/O, 12 I/O, dan 20 I/O.
4. Memiliki slot yang dapat digunakan untuk slot memory, koneksi ke PC (*Personal Computer*) serta koneksi untuk *Interface* komunikasi.
5. Display LCD 4 baris dengan 18 karakter.
6. Baterai yang digunakan untuk mengoperasikan waktu pada Zelio Logic *Smart Relay* terbuat dari bahan *lithium battery* yang dapat bertahan hingga 10 tahun. Data *Backup* menggunakan sebuah EEPROM *Flash Memory* yang dapat bertahan hingga 10 tahun.
7. Apabila dibutuhkan, Zelio Logic *Smart Relay* dapat menggunakan I/O *Extension* :
6, 10 or 14 I/O, *supplied with a 24 VDC via smart relay*,

6, 10 or 14 I/O, *supplied with a 24 VAC via smart relay*

6, 10 or 14 I/O, *supplied with a 100 - 240 VAC via smart relay.*

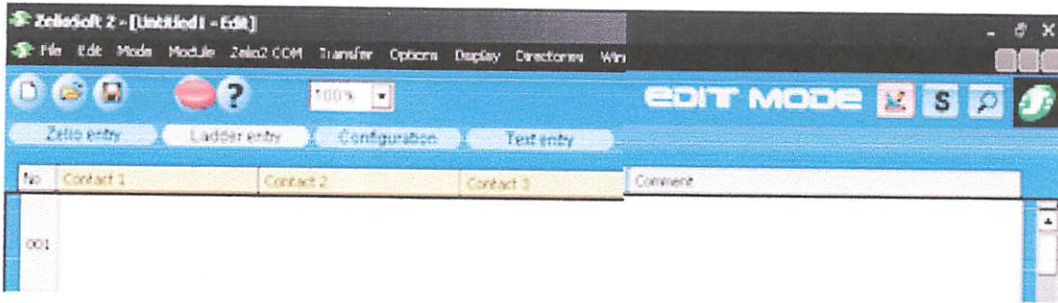
8. Modul komunikasi untuk *Modbus Network* tersedia bagi *Modular Smart Relay* yang disuplai dengan tegangan 24 VDC. Sedangkan *Communication Interface* digunakan untuk mengkomunikasikan antara *Zelio Logic Smart Relay* dan PC. *Communication Interface* didesain untuk *me-monitoring* atau digunakan sebagai *Remote Control* bagi mesin ataupun instalasi dimana operasi sistemnya tidak menggunakan operator.

2.2.4.2 Zelio Soft Software ^[3]

Zelio Soft Software memungkinkan untuk :

1. *Programming* menggunakan Ladder dan *Function Block Diagram (FBD)*.
2. *Simulasi, Monitoring, Supervision dan Programming.*
3. *Uploading dan Downloading Program.*
4. *Meng-compile program secara otomatis.*
5. *On-line Help.*

Software Soft dapat memonitor program itu sendiri dalam arti apabila terjadi kesalahan dalam pemrograman, maka akan muncul indikator berwarna merah yang menandakan bahwa sistem *error*. Problem dapat diketahui dengan meng-*click* mouse pada indikator yang berwarna merah.



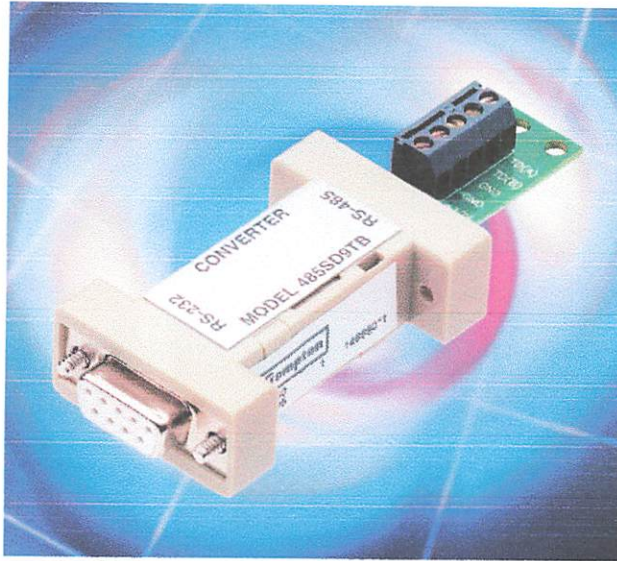
Gambar 2..12
Indikator kesalahan pemrograman pada Zelio Soft

Pada Zelio Soft terdapat dua tipe mode pengetesan, yaitu :

1. Zelio Soft *Simulation mode* yang memungkinkan program dapat diuji tanpa menggunakan Zelio Logic Smart Relay. Antara lain dapat menampilkan :
 - Menampilkan status dari *output*.
 - Dapat bervariasi tegangan dari *analog input*.
 - Memungkinkan untuk pemrograman tombol tau *buttons*.
 - Simulasikan sistem dengan menggunakan fasilitas *Real Time Clock* (RTC).
 - Tampilan sangat dinamik (berwarna merah) pada variable elemen yang aktif dari pemrograman.
2. Zelio Soft *Monitoring mode* yang memungkinkan program dapat di-*test* bersama – sama dengan Zelio Logic *Smart Relay* yang dapat :
 - Menampilkan program secara *on-line*.
 - Mengatur waktu.
 - Memonitoring sistem melalui *hardware* dan *software* secara bersamaan.

- Mengoptimalkan *input*, *output*, *control relay* dan *current values* dari *function block*.
- Dapat mengganti dari *stop* dan *run mode*.

Untuk komunikasi antara *Modbus* dengan PC/Laptop digunakan kabel D-485/9D RS232 to RS485 Converter.



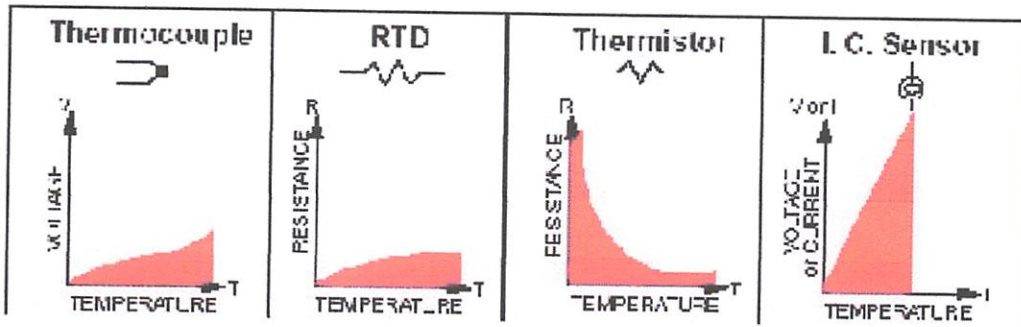
Gambar 2.13
Kabel D-485/9D RS232 to RS485 Converter.^[8]

2.3 Sensor Suhu^[10]

Prinsip kerja sensor suhu ini adalah dengan mengatur serta menstabilkan suhu dalam ruangan inkubator agar sesuai dengan suhu yang dibutuhkan oleh bayi premature. Alat ini menggunakan pemanasan elemen (*heater*) yang dikontrol oleh suatu rangkaian kontrol suhu agar suhu tetap stabil. *Heater* akan bekerja pada saat sensor suhu kurang dari setting suhu yang telah ditentukan, dan sebaliknya apabila sensor suhu lebih besar dari setting suhu, secara otomatis heater akan mati dan kipas (*fan*) akan bekerja.

2.3.1 Sensor Suhu (IC LM 35) ^[10]

Berdasarkan karekteristik tiap sensor yang berada dipasaran dan keunggulan masing – masing sensor yang dapat dilihat pada gambar dibawah ini, maka penggunaan IC Sensor adalah menjadi pilihan utama.



Gambar 2.14
Karakteristik Sensor Suhu ^[10]

Ada beberapa IC Sensor yang dapat digunakan untuk mengukur suhu. Setelah membandingkan aplikasi dan spesifikasi masing – masing IC Sensor maka IC Sensor LM 35 adalah yang paling tepat.

IC LM 35 sebagai sensor suhu yang teliti dan terkemas dalam bentuk *Integrated Circuit* (IC) 3 pin yang berbentuk seperti transistor FCS9013, dimana *output* tegangan keluaran sangat linear terhadap perubahan suhu. Sensor ini berfungsi sebagai pengubah dari besaran fisis suhu ke besaran tegangan yang memiliki koefisien sebesar $10 \text{ mV}/^\circ\text{C}$ yang berarti bahwa kenaikan suhu 1°C maka akan terjadi kenaikan tegangan sebesar 10 mV .

IC LM 35 ini tidak memerlukan pengkalibrasian atau penyetelan dari luar karena ketelitiannya sampai lebih kurang seperempat derajat celcius pada temperature ruang. Jangka sensor mulai dari -55°C sampai dengan 150°C , IC LM35 penggunaannya sangat mudah, difungsikan sebagai kontrol dari indikator

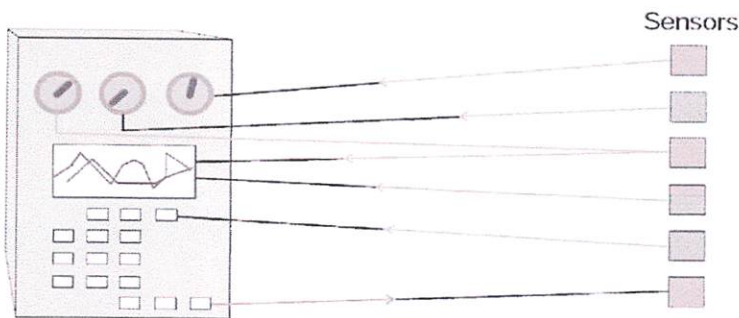
Control = pengendalian

Data acquisition = akuisisi data (pengumpulan/pengambilan data)

SCADA merupakan sebuah sistem yang mengumpulkan informasi atau data-data dari lapangan dan kemudian mengirimkannya ke sebuah komputer pusat yang akan mengatur dan mengontrol data-data tersebut.

Primitive SCADA

Pada awalnya sistem SCADA menggunakan panel meter, lampu dan chart recorder untuk keperluan data akuisisi. Operator secara manual mengoperasikan tombol (*knob*) untuk melakukan pengendalian. Gambar 2.15 memperlihatkan sebuah sistem *sensor to panel* untuk *implementasi primitive* SCADA ini.



Gambar 2.16

Sistem "Sensor to panel" menggunakan arus 4-20 mA atau tegangan. [2]



Gambar 2.17

Panel "Primitive" SCADA [2]

Modern SCADA

Pada modern SCADA, komputer digunakan sebagai pengganti panel seperti terlihat pada gambar 2.18.

SCADA pada dasarnya merupakan kombinasi dari *telemetry* dan *data acquisition*.

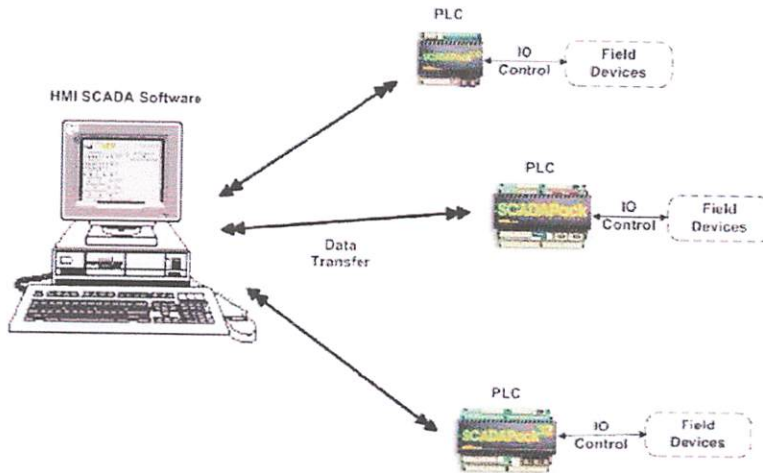
Telemetry merupakan sistem yang digunakan untuk mengirim perintah dan menerima/monitor data dari jarak jauh.

Data acquisition merupakan metode yang digunakan untuk mengakses informasi dari peralatan yang akan dikendalikan atau dimonitor.

SCADA mengumpulkan informasi dari lapangan, selanjutnya dikirimkan ke pusat untuk diolah dan dianalisa yang informasinya ditampilkan di layar komputer operator, serta akan mengirimkan sinyal kontrol ke lapangan. Tipikal sistem SCADA diperlihatkan pada gambar 2.19.



Gambar 2.18
Komputer sebagai panel monitoring dan pengendalian pada SCADA^[2]



Gambar 2.19
Tipikal Sistem SCADA [2]

2.4.1.2 Komponen SCADA [2]

Komponen dari sistem SCADA adalah:

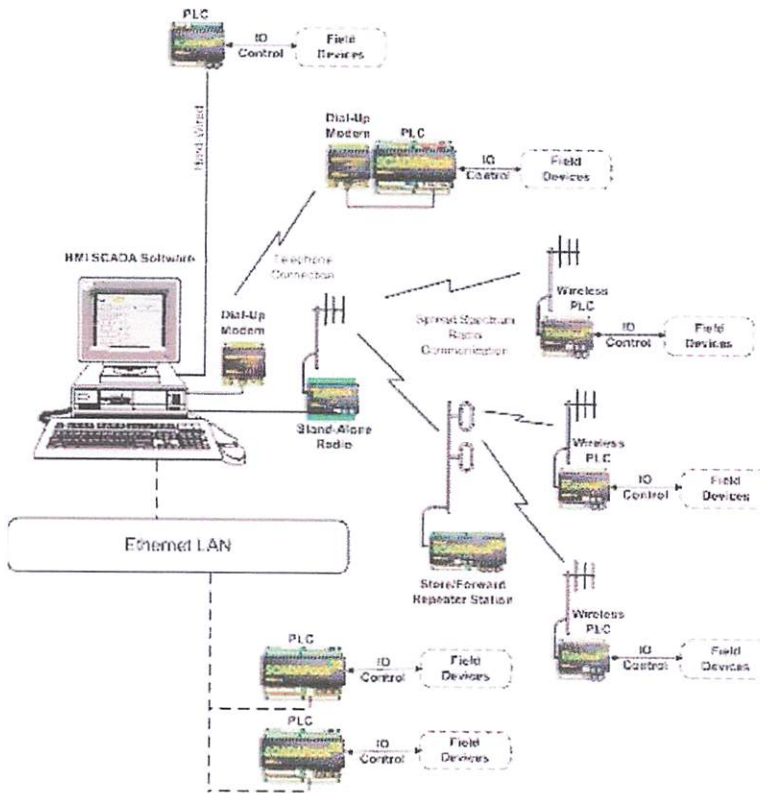
1. Peralatan lapangan (sensor dan aktuator) : Peralatan *instrumentasi* di lapangan berupa sensor digunakan untuk membaca sinyal analog atau digital yang diukur, sedangkan aktuator digunakan untuk mengendalikan peralatan seperti motor, saklar, katup, dll.
2. *Remote terminal unit* (RTU) : RTU atau remote station umumnya berupa PLC (*Programmable Logic Controller*) yang berfungsi untuk mengendalikan aktuator, membaca sinyal dari sensor dan berkomunikasi dengan pusat pengendali.
3. Jaringan komunikasi : digunakan untuk menghubungkan RTU dengan stasiun pusat pengendali yang berupa jaringan kabel, atau radio.
4. Stasiun pusat pengendali : merupakan stasiun pusat monitoring dan pengendalian dimana terdapat komputer host dengan perangkat lunak

SCADA/HMI (*Human Machine Interface*) yang memudahkan operator melakukan monitoring dan pengendalian peralatan-peralatan di lapangan.

2.4.1.2.1 Sistem Komunikasi SCADA ^[2]

SCADA dapat menggunakan beberapa jenis komunikasi seperti terlihat pada gambar 2.20 :

- *Serial RS232*
- *Private Network (LAN/RS-485)*
- Jaringan Telepon (PSTN)
- *Leased line*
- Internet
- Sistem komunikasi *wireless* (Wireless LAN, GSM, Radio modem)



Gambar 2.20
Sistem Komunikasi pada SCADA ^[2]

2.4.1.2.2 Software SCADA ^[2]

Software SCADA dapat dibagi menjadi dua tipe : Proprietary dan Open. Pada tipe proprietary software, perusahaan menjualnya untuk berkomunikasi dengan peralatan yang mereka miliki. Masalah utama dengan sistem ini adalah ketergantungan pada perusahaan penyedia software dan hardware.

Sistem dengan Open Software lebih *popular* karena sifatnya yang *interoperability*, yaitu kemampuan untuk menggabungkan peralatan dari berbagai merek/perusahaan dalam suatu sistem. Contoh dari Open Software yang dijual di pasaran adalah *Citect* dan *Wonderware*.

Fitur dari Software SCADA adalah :

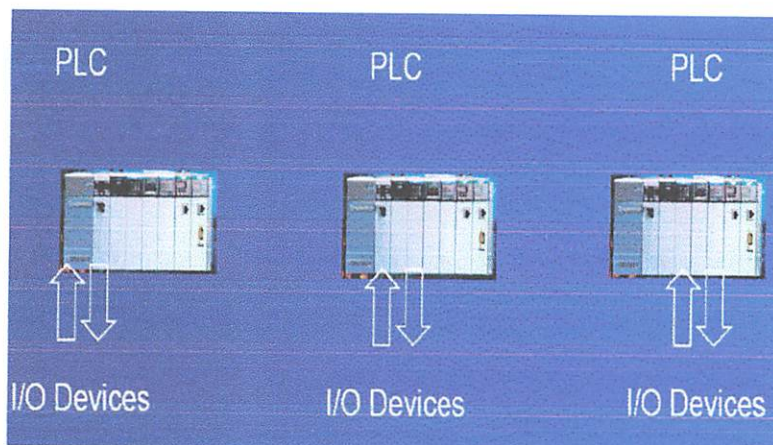
- *User Interface* :
 - o *Keyboard*
 - o *Mouse*
 - o *Trackball*
 - o *Touch Screen*
- *Graphic display* :
 - o Resolusi tinggi
 - o Dapat dikonfigurasi oleh user
- Alarm (fasilitas alarm)
 - o Alarm ditampilkan dalam urutan kronologis
 - o Format dan warna dapat didefinisikan oleh user
 - o *Historical alarm* dan *event logging*
 - o *Event-triggered alarm*
 - o *Alarms-triggered report*

- Trends:
 - *Trend* dapat diekspor ke file DBF, CSV
 - Dapat diplot dalam sumbu X/Y
 - *Event based trends*
 - Ditampilkan secara pop-up
 - *Short term* dan *long term* trend
- RTU (dan PLC):
 - Didukung oleh *protokol* standar
 - Tersedia *Driver toolkit*
 - Beroperasi berbasis permintaan, sebagai pengganti metode scan
- *Scalability*:
 - Penambahan *hardware* dapat dilakukan tanpa mengganti atau memodifikasi peralatan yang ada.
- Access to data:
 - Data dapat diakses secara langsung dan *real time*
 - *Real time* data dapat dibaca oleh program lain, misalnya *Excel*
- Database:
 - Didukung oleh ODBC *driver*
 - Didukung oleh perintah SQL
- Networking:
 - Didukung oleh jaringan seperti, *Netware, Windows for Workgroup, Windows NT*
 - Didukung oleh *protocol* NetBEUI, TCP/IP
- Fault tolerance dan redundancy

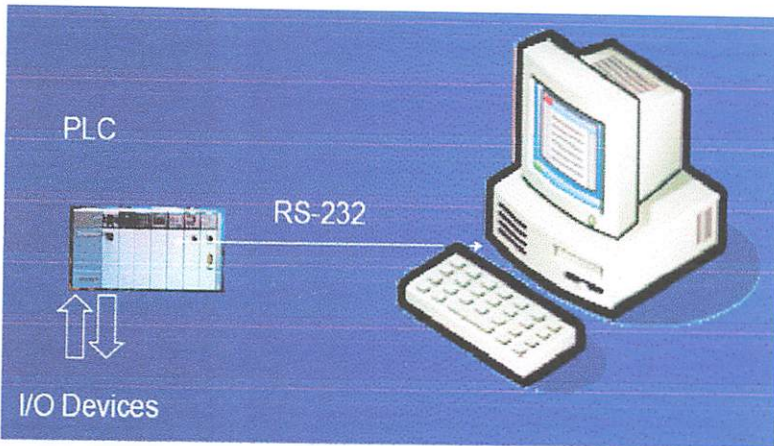
- *Redundansi* dapat diterapkan pada *hardware* tertentu
- *Redundansi* pada jaringan/LAN
- *Client/server distributed processing*
 - *Real time multitasking*

2.4.1.3 Perkembangan SCADA ^[2]

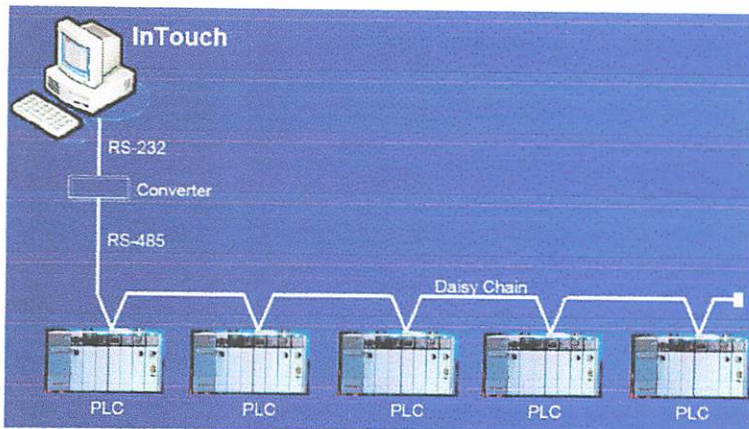
- Stand alone PLC
- PLC dengan HMI (*Human Machine Interface*)
- Multiple PLC dengan HMI (*RS-485 Multidrop*)
- Multiple PLC dengan HMI (*Special Bus*)
- Multiple PLC dengan Multiple HMI (*Special Bus* dan TCP/IP)
- Multiple PLC dengan Multiple HMI (TCP/IP)



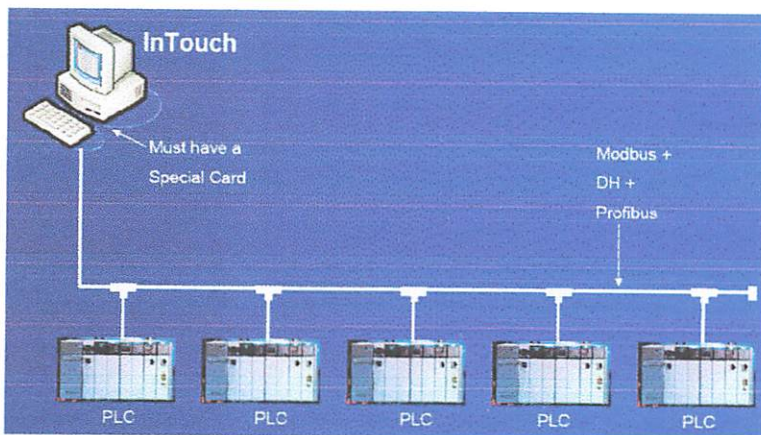
Gambar 2.21
Stand alone PLC ^[2]



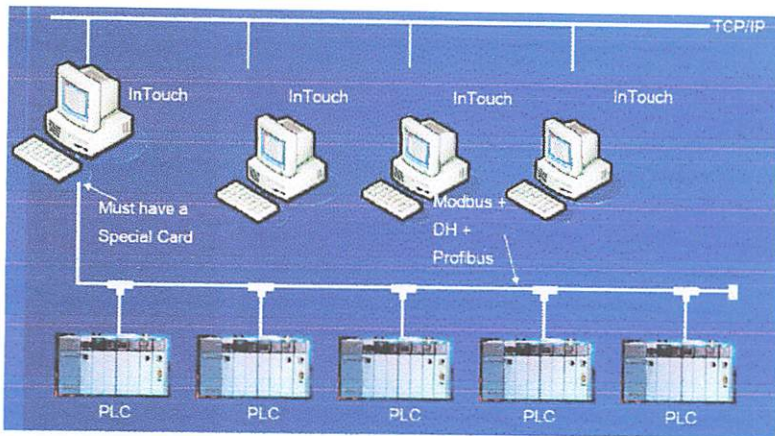
Gambar 2.22
PLC dengan HMI ^[2]



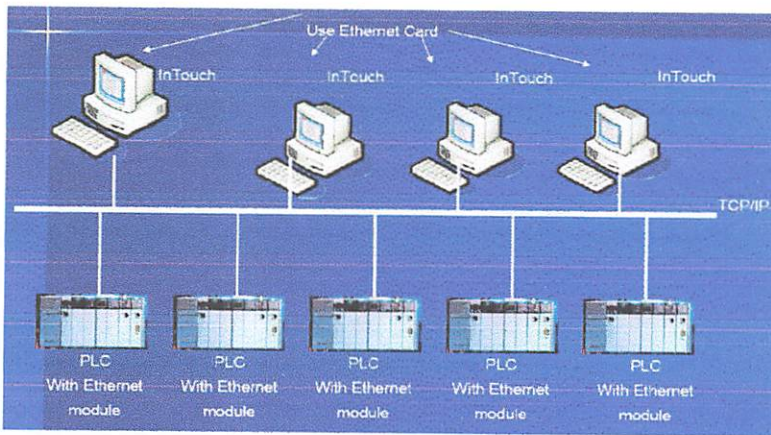
Gambar 2.23
Multiple PLC dengan HMI (RS-485 Multidrop) ^[2]



Gambar 2.24
Multiple PLC dengan HMI (Special Bus) ^[2]

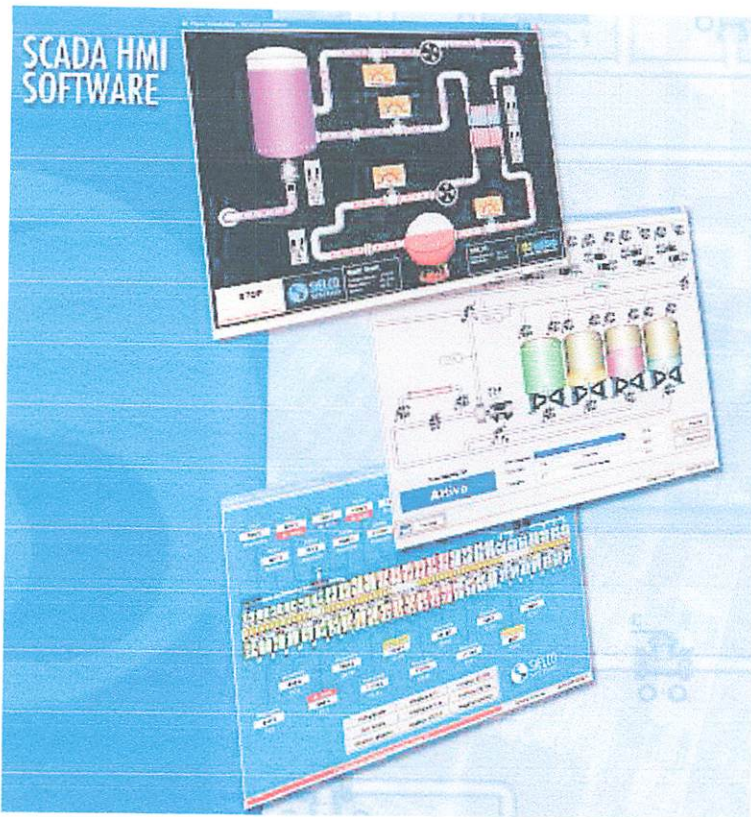


Gambar 2.25
Multiple PLC dengan Multiple HMI (Special Bus dan TCP/IP) [2]



Gambar 2.26
Multiple PLC Multiple HMI (TCP/IP) [2]

2.4.2. WINLOG SCADA Lite ^[2]



2.4.2.1 Tentang Winlog ^[2]

- Winlog adalah paket perangkat lunak SCADA/HMI yang sederhana dan fleksibel untuk supervisi proses industri.
- Pada Winlog terdapat tool-tool seperti *Gate Builder*, *Template Builder*, *Code Builder* untuk mempermudah pembuatan aplikasi SCADA.
 - o *Gate Builder* : digunakan untuk *setting* gate (tags), seperti definisi nama, alamat, unit pengukuran, faktor skala,dll. Terdapat berbagai macam gate, yaitu : *numeric*, *digital*, *string*, *compound*, *even*, *alarm*. Gate merupakan variabel proses yang dapat dibaca dari peralatan luar, seperti PLC, indikator, modul data *akuisisi*, atau dihasilkan oleh program sendiri.

- Template builder : digunakan untuk membuat template dan menampilkannya di layar. Sebuah template digunakan untuk menyusun objek yang digunakan, seperti gambar bitmap, teks, nilai, status bar, led indikator, dll.
- Code builder : digunakan untuk memperkaya aplikasi menggunakan bahasa pemrograman seperti bahasa C yang memungkinkan programmer berinteraksi dengan semua elemen Winlog, seperti membuat “looping”, atau kondisi “if-then-else”
- Terdapat bermacam-macam *library driver* yang memungkinkan Winlog berkomunikasi dengan berbagai jenis/merek PLC.

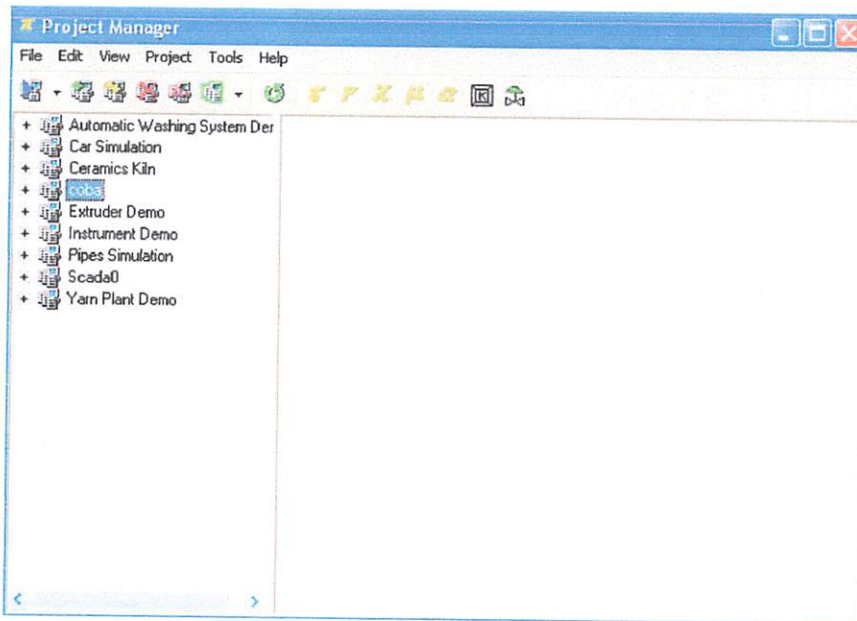
Winlog Pro vs Winlog Lite

- Winlog Pro merupakan versi lengkap dan tidak *free*.
- Winlog Lite merupakan “*entry level*” dari versi Winlog Pro
- Winlog Lite menyediakan semua *development tool* dan fungsi yang ada pada Winlog Pro, hanya dibatasi sampai maksimum 24 tag.
- Winlog Lite dapat dijalankan dalam dua mode : *Demo mode dan Full mode*. Pada demo mode, komunikasi dengan piranti luar (PLC) akan berhenti secara otomatis setelah 15 menit (jika diinginkan, dapat di-restart kembali secara manual) ; pada Full Mode, komunikasi dengan PLC dapat berlangsung tanpa batasan waktu.
- Untuk menjalankan Winlog Lite dalam Full mode, diperlukan “*license*” dan proses registrasi

2.4.2.2 Panduan Menggunakan Winlog ^[2]

2.4.2.2.1. Project Manager ^[2]

Project manager digunakan untuk membuat dan mengelola berbagai macam *project* SCADA. Ketika *project* manager dijalankan, maka akan tampil *project-project* yang sudah ada/dibuat. Selanjutnya kita dapat membuat, menghapus, meng-copy, mengkonfigurasi *project*. Gambar 2.27 memperlihatkan tampilan *Project Manager*, dimana di sebelah kiri terdapat nama-nama *project* yang ada.

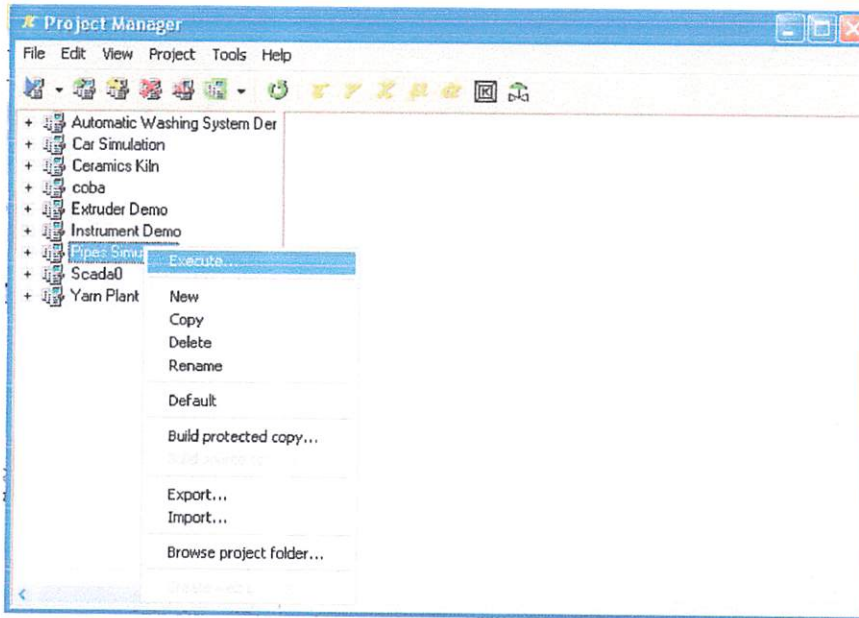


Gambar 2.27
Tampilan Project Manager ^[2]

2.4.2.2.2 Pengoperasian Project ^[2]

2.4.2.2.2.1 Menjalankan project

Pilih nama *project* yang akan dieksekusi/dijalankan, tekan tombol kanan mouse dan pilih *Execute* dari menu di layar, atau pilih *Execute* dari *Project* Menu (lihat gambar 2.28).



Gambar 2.28
Menjalankan project. [2]

2.4.2.2.2.2 Membuat project baru

Pilih *Project|New*, atau klik kanan *mouse*, dan pilih *New*. Selanjutnya akan ditanyakan nama *Project* yang akan dibuat.

2.4.2.2.2.3 Meng-copy project

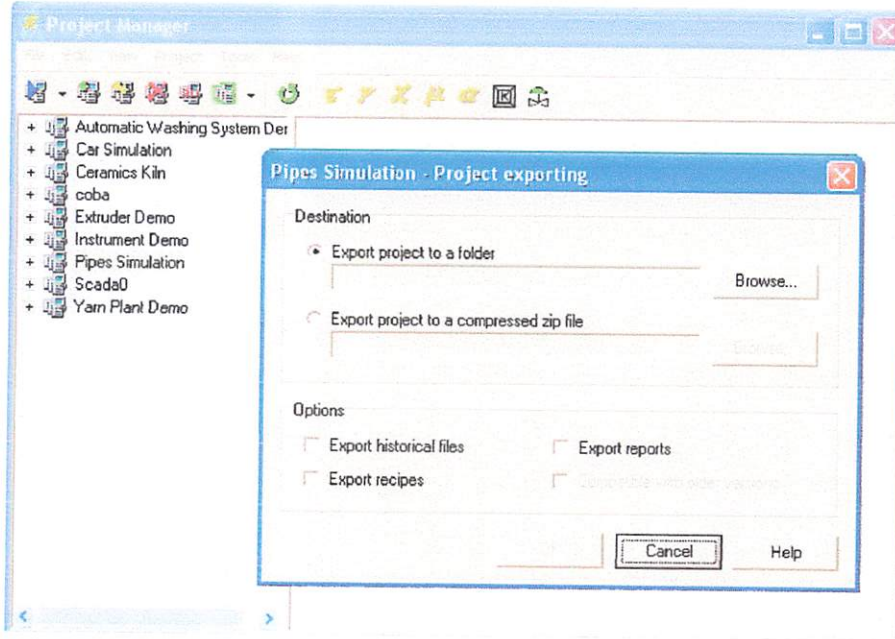
Pilih nama *project* yang akan di-copy, lalu Pilih *Project|Copy* atau klik kanan *mouse*, dan pilih *Copy*. Selanjutnya akan ditanyakan nama *project* hasil peng-copy-an.

2.4.2.2.2.4 Menghapus project

Pilih nama *project* yang akan dihapus, dan *Delete* dari *project* menu, atau dari klik kanan *mouse* lalu *Delete*.

2.4.2.2.2.5 Mengekspor File

Digunakan untuk menyimpan/ekspor project ke direktori selain direktori *Project* dari Winlog. Pilih nama *project* yang akan diekspor, lalu pilih *Project|Export*, atau klik kanan mouse, lalu *Export* (lihat gambar 2.29)



Gambar 2.29
Mengekspor file ^[2]

2.4.2.2.2.6 Mengimpor File

Digunakan untuk mengimpor/mengambil *project* dari direktori lain. Pilih *Project|Import*, atau klik kanan mouse, lalu *Import*.

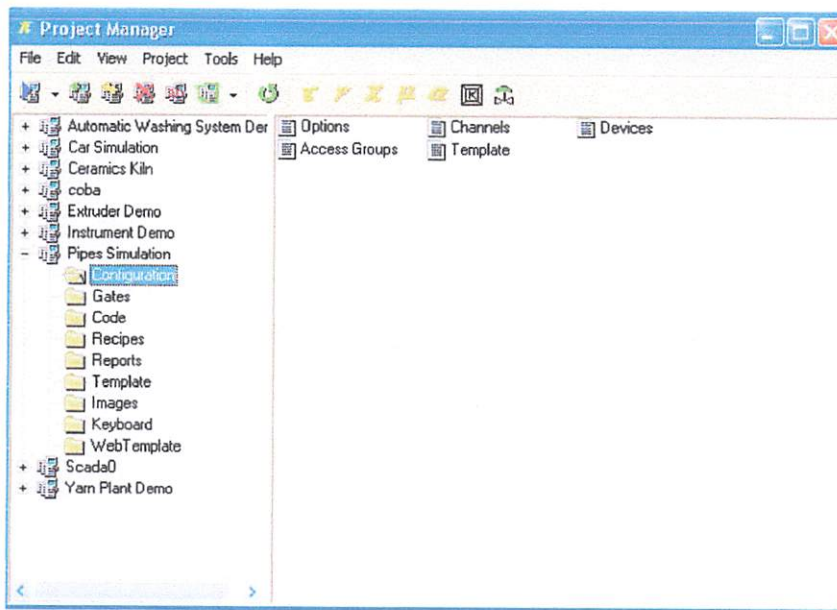
2.4.2.2.2.7 Membuka *windows* folder dari *project-project* yang ada

Pilih *Project|Browse project folder*, atau klik kanan mouse, lalu *Browse project folder*.

2.4.2.2.3 Konfigurasi ^[2]

Digunakan untuk mengkonfigurasi *project*. Untuk masuk menu konfigurasi, pilih nama *project*, lalu klik tanda + di sebelah kiri nama *project* untuk melihat sub-menu di dalamnya. Selanjutnya pilih *Configuration*.

Dalam menu *Configuration* ini, terdapat sub-menu : *Options*, *Channels*, *Devices*, *Access Groups*, dan *Template* (lihat gambar 2.30)



Gambar 2.30
Menu Configuration ^[2]

2.4.2.2.3.1 Options

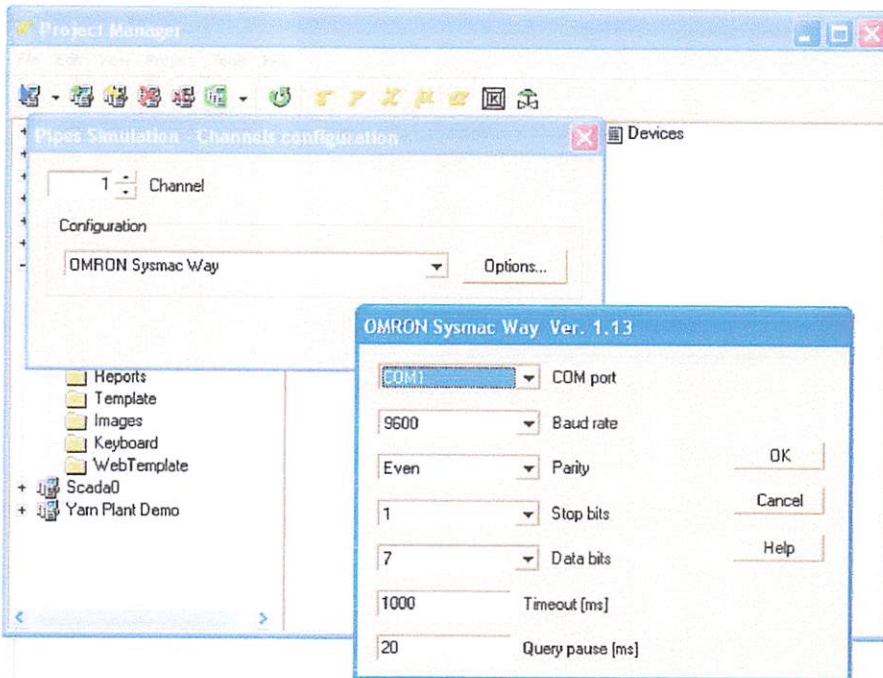
Digunakan untuk setting fitur *project*, seperti judul *project*, *deskripsi project*, tampilan *windows* Runtime saat dijalankan, *file history*, alamat TCP/IP untuk konfigurasi Client-server, dll.

2.4.2.2.3.2 Channel

Merupakan *port* komunikasi yang digunakan untuk berkomunikasi dengan piranti luar.

Langkah-langkah untuk konfigurasi *channel* adalah (lihat gambar 2.31):

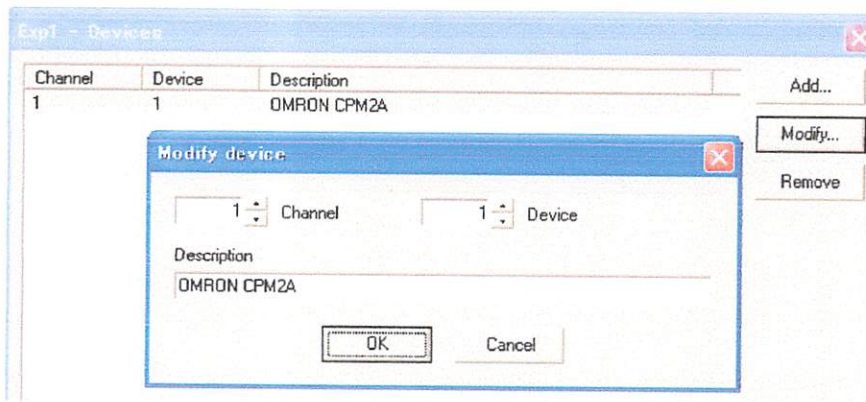
- Pilih nomer *channel*
- Pilih konfigurasi *protkcol* yang disediakan
- Tentukan Com port (Serial Port) dari PC yang digunakan
- Set parameter komunikasi yang sesuai dengan piranti luar/PLC yang digunakan.



Gambar 2.31
Konfigurasi Channel. [2]

2.4.2.2.3.3 Devices

Winlog SCADA dapat memonitor komunikasi dari semua peralatan yang terkoneksi ke komputer. Komunikasi diidentifikasi oleh nomer *channel* dan alamat seperti terlihat pada gambar 2.32.



Gambar 2.32
Konfigurasi Device ^[2]

2.4.2.2.4 Gate

Gate digunakan untuk menyatakan suatu variabel proses.

Terdapat lima tipe gate, yaitu:

- *Numerical* : bernilai *floating point*
- *Digital* : bernilai 0 atau 1
- *String* : berupa deretan karakter (maksimum 80 karakter)
- *Composite/Compound* : nilainya merupakan hasil operasi dari dua gate
- *Event/Alarm* : aktif tergantung nilai dari gate tertentu

Untuk melakukan konfigurasi gate dapat dilakukan dengan *double klik* pada icon gate, dan selanjutnya akan tampil Gate Builder.

2.4.2.2.5 Code

Digunakan untuk menuliskan code/program yang akan dijalankan. Untuk membuat file baru (bila belum ada) dilakukan dengan klik kanan mouse dan pilih *New*. Selain itu dapat juga dilakukan operasi file umum (*Copy, cut, paste, rename*).

Untuk mengakses file dilakukan dengan klik ganda pada ikonnya, dan selanjutnya akan tampil Code Builder.

2.4.2.2.6 Template

Merupakan halaman yang digunakan untuk proses supervision. Seperti halnya Code, kita dapat membuat file baru, *copy*, *cut*, *paste* dengan klik kanan mouse.

Untuk mengakses file dilakukan dengan klik ganda pada ikonnya, dan selanjutnya akan tampil Template Builder.

2.4.2.3. Gate Builder

Digunakan untuk mengkonfigurasi gate yang dipakai untuk supervisi suatu proses.

Terdapat lima macam gate yang dapat digunakan (numerical, *digital*, string, compound, dan event/alarm). Gate *builder* dapat dijalankan dari menu *Tools|Gate*

Builder atau dari klik kanan mouse pada icon Gates di bawah nama project, dan

klik ganda pada tipe gates yang diinginkan. Menu tampilan Gate Builder dapat

dilihat pada gambar 2.33.

	Channel	Device	Gate ID	N ID	Address	Description	Measure	Variable type	Tolerance	Min. value	Max. value	Start value
1		0	Temperature	1				DOUBLE	0	0	0	
2	1	0	Flow	1				DOUBLE	0	0	0	
3	1	0	Temperature	2				DOUBLE	0	0	0	
4	1	0	Flow	2				DOUBLE	0	0	0	
5	0	0	Level	1				DOUBLE	0	0	80	
6	0	0	Status	1		1= cold 2= warm 3= hot 4= off		DOUBLE	0	0	2	
7	0	0	Status	2		1= cold 2= warm 3= hot 4= off		DOUBLE	0	0	3	

Gambar 2.33
Gate Builder. [2]

Untuk menambah atau mengedit gate dilakukan dengan klik ganda *mouse*, atau klik kanan mouse, lalu pilih *Insert* atau *Edit*.

Untuk mengecek kesalahan konfigurasi gate digunakan menu *File|Error Checking*.

Beberapa hal yang harus diperhatikan dalam konfigurasi gate:

- Cek apakah *sampled gate* mempunyai *protocol komunikasi*
- Cek apakah alamat dari setiap *sampled gate* sudah ditulis dengan benar sesuai dengan *channel* yang dipilih untuk gate tersebut
- Cek apakah terdapat *samped gates* pada *channel 0* atau *device 0* (Pada *Winlog*, *Channel 0* dan *device 0* tidak dapat digunakan untuk mengakses peralatan eksternal).

2.4.2.3.1 Numerical gate

Setting parameter pada *numerical gate* terdiri dari empat halaman/Tab :

- *General* : untuk mendefinisikan data umum suatu gate
- *Sampling* : untuk mendefinisikan data yang berhubungan dengan *sampling* suatu gate
- *Value* : untuk mendefinisikan bagaimana nilai dibaca dari gate
- *Tolerance* : toleransi data yang digunakan

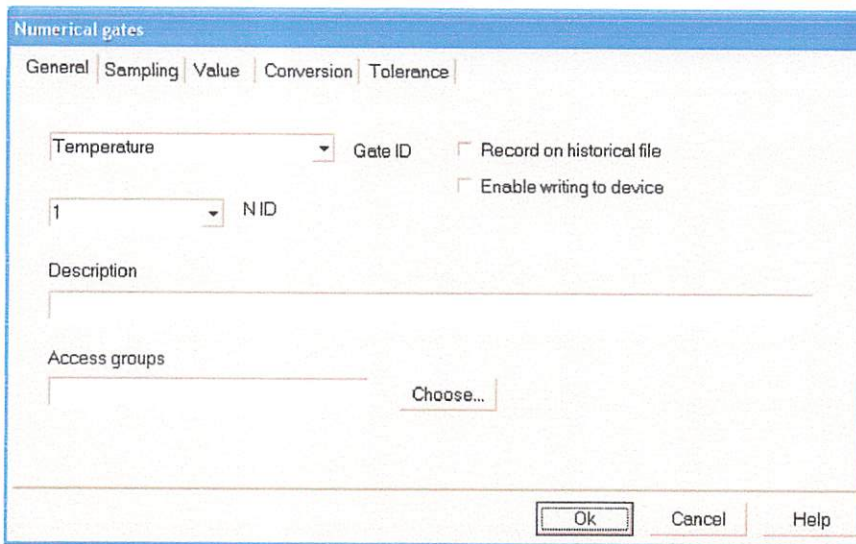
2.4.2.3.1.1 General

Pada halaman *General*, terdapat beberapa parameter (lihat gambar 2.34):

- Gate ID dan N ID: untuk identifikasi gate di *project*, gate ID dan N ID ini harus unik (setiap gate mempunyai gate ID dan N ID yang berbeda dengan

gate lainnya). Gate ID dapat diisi dengan karakter (maksimum 20 karakter), N ID harus berupa angka 0 s/d 65535.

- *Record on historical file* : jika pilihan ini diset, maka variasi data yang diperoleh dari peralatan luar dan dibaca oleh Gate akan disimpan pada file history.
- *Enabling writing to device* : jika diset, maka perubahan nilai pada gate akan dikirimkan ke peralatan luar
- *Description* : diisi keterangan dari suatu gate (maksimum 80 karakter)
- *Access groups* : digunakan untuk mengindikasikan user group yang mana yang diperbolehkan mengubah parameter gate.



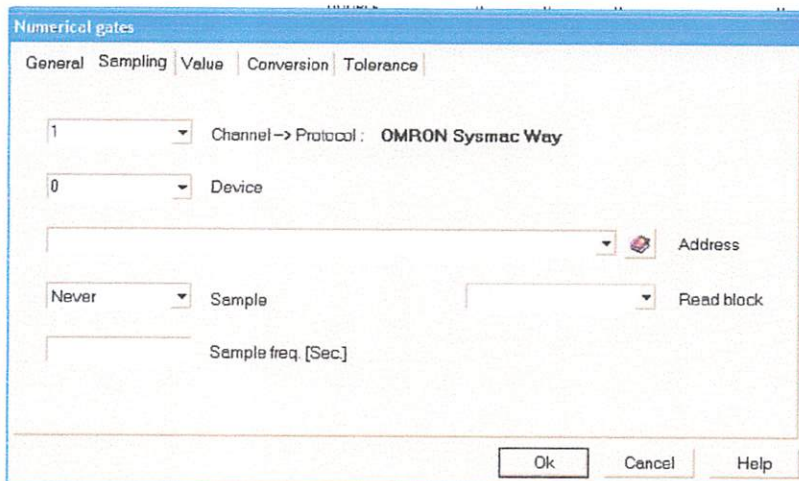
Gambar 2.34
Parameter General pada Numerical gate. [2]

2.4.2.3.1.2 *Sampling*

Parameter *Sampling* terdiri dari (lihat gambar 2.35).

- *Channel*: menandakan channel (yang didefinisikan pada bagian configuration) yang digunakan untuk men-sampling data.

- *Device* : nomer *device* (yang didefinisikan pada bagian configuration) dimana nilai diambil/sampling
- *Address* : menunjukkan tipe dan nomer register dari data yang dibaca/ditulis ke peralatan. (Setiap protokol mempunyai aturan yang berbeda-beda mengenai cara penulisan address ini)
- *Sample* : Terdapat tiga pilihan, yaitu
 - *Never* : gate tidak pernah disampling dari peralatan luar
 - *Always* : gate selalu disampling
 - *If in monitor* : gate disampling hanya jika diperlukan
- *Sample freq* : menyatakan interval waktu pen-samplingan (dalam detik)
- *Read block* : untuk mendefinisikan blok dari sekumpulan gate, untuk mempercepat proses pembacaan dari alat.



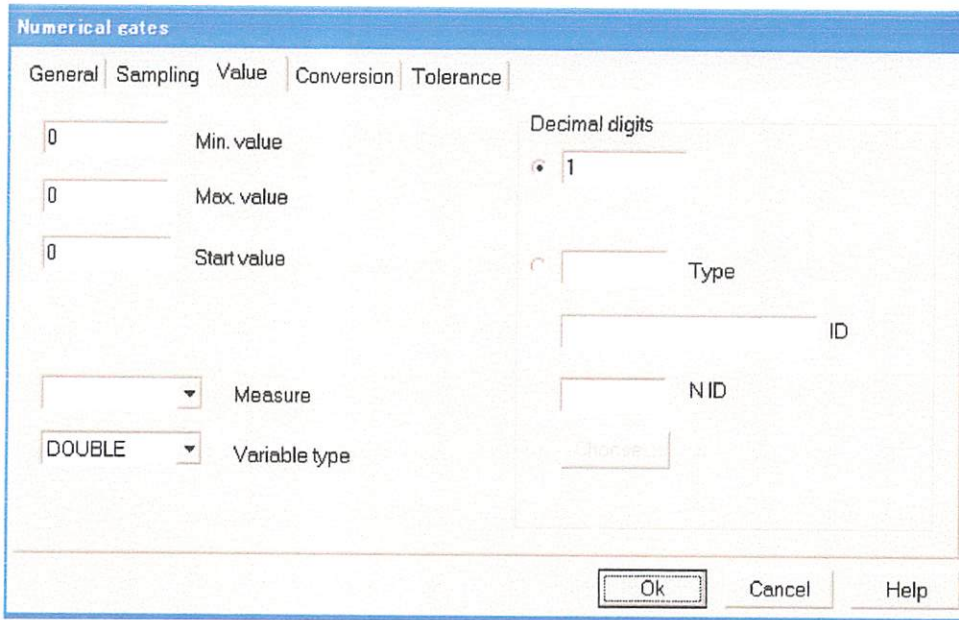
Gambar 2.35
Parameter Sampling pada Numerical gate. ^[2]

2.4.2.3.1.3 Value

Parameter value terdiri dari (lihat gambar 2.36):

- *Min. value, Max. value*: digunakan untuk membatasi nilai gate

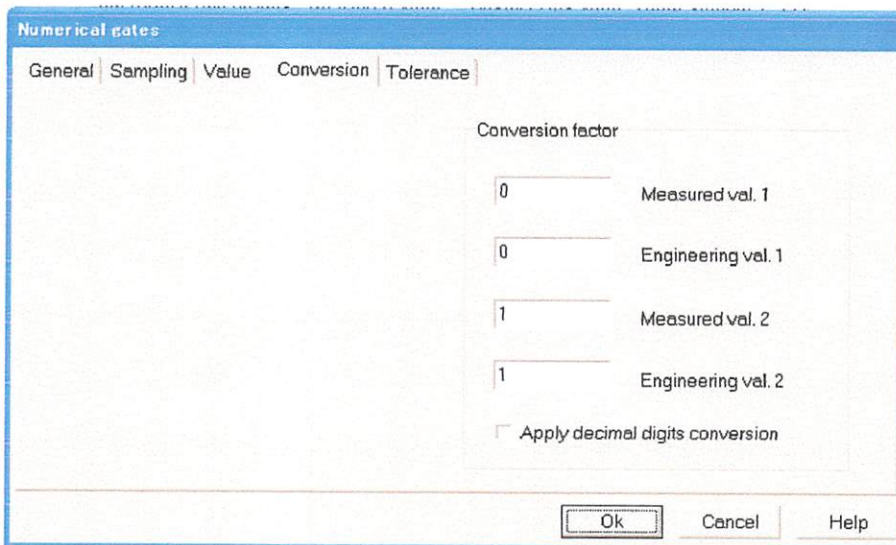
- *Start value* : nilai yang diberikan pada gate pada saat awal aplikasi dijalankan
- *Measure* : menyatakan satuan pengukuran (hanya sebagai informasi, tidak berpengaruh pada nilai)
- *Variable type* : tipe variabel gate yang digunakan
 - S_BYTE : satu byte signed integer
 - U_BYTE : satu byte unsigned integer
 - S_WORD : dua byte signed integer
 - U_WORD : dua byte unsigned integer
 - S_INT32 : empat byte signed integer
 - U_INT32 : empat byte unsigned integer
 - FLOAT : floating point
 - DOUBLE : floating point double precision
 - BCD : dua byte unsigned Binary Coded Decimal
- *Decimal digits* : jumlah desimal yang diinginkan dari hasil nilai pengukuran



Gambar 2.36
Parameter value pada Numerical gate ^[2]

2.4.2.3.1.4 Conversion

Conversion factor: digunakan sebagai faktor konversi dari nilai yang dibaca melalui *interpolasi* dua pasang *Measured value* – *Engineering value* (lihat gambar 2.37).

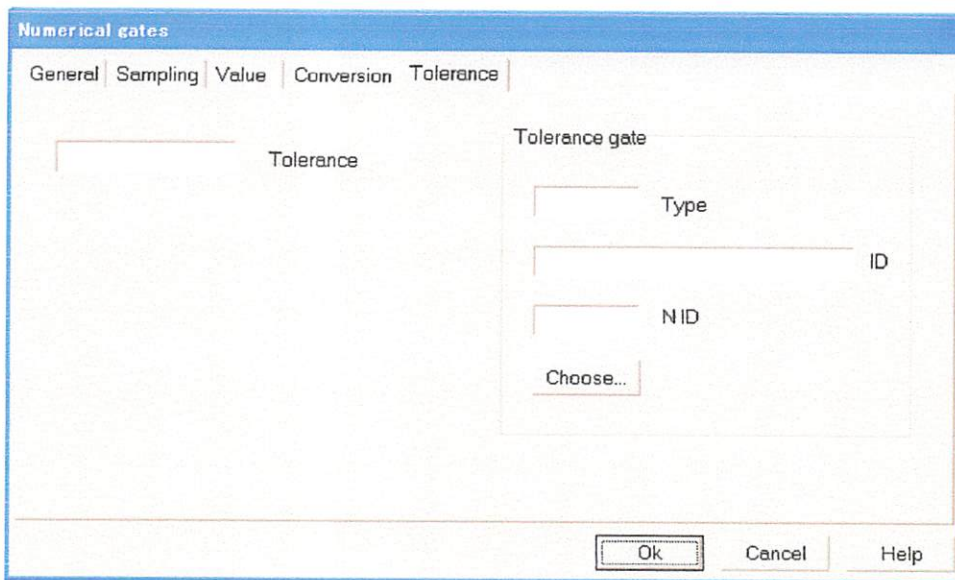


Gambar 2.37
Parameter *Conversion* pada *Numerical Gate*. ^[2]

2.4.2.3.1.5 Tolerance

Parameter *tolerance* terdiri dari *Tolerance* dan *tolerance gate* (lihat gambar 2.38):

- *Tolerance* : nilai toleransi pada gate. Nilai toleransi digunakan untuk memfilter data yang ditampilkan pada halaman supervisi, untuk menghindari nilai yang berubah-ubah, dan memungkinkan untuk tidak menyimpan nilai – nilai yang hampir sama. Sebagai contoh, jika nilai *tolerance* adalah 3, maka jika nilai gate 20 dan nilai yang dibaca oleh peralatan 22, maka nilai gate tidak akan berubah, nilai gate akan berubah jika nilai yang dibaca lebih dari atau sama dengan 23.
- *Tolerance gate* : digunakan untuk membatalkan nilai toleransi jika nilai gate sama dengan nilai *tolerance gate*.



Gambar 2.38
Parameter Tolerance pada Numerical gate. ^[2]

2.4.2.3.2 Digital gates ^[2]

Pada digital gate, terdapat tiga halaman untuk *setting* parameter :

- General : sama dengan pada numerical gate

- *Sampling* : sama dengan pada *numerical gate*
- *Value* : hanya berisi *start value* : nilai *gate* pada saat aplikasi mulai dijalankan.

2.4.2.3.3 *String gate* ^[2]

Sama dengan *digital gate*, pada *string gate* juga terdapat tiga halaman setting:

- *General* : sama dengan pada *digital gate*
- *Sampling* : sama dengan pada *digital gate*
- *Value* : berisi *Max dimension* yang menyatakan panjang string maksimum dan *start value* yang menyatakan nilai string dari *gate* saat awal dijalankan.

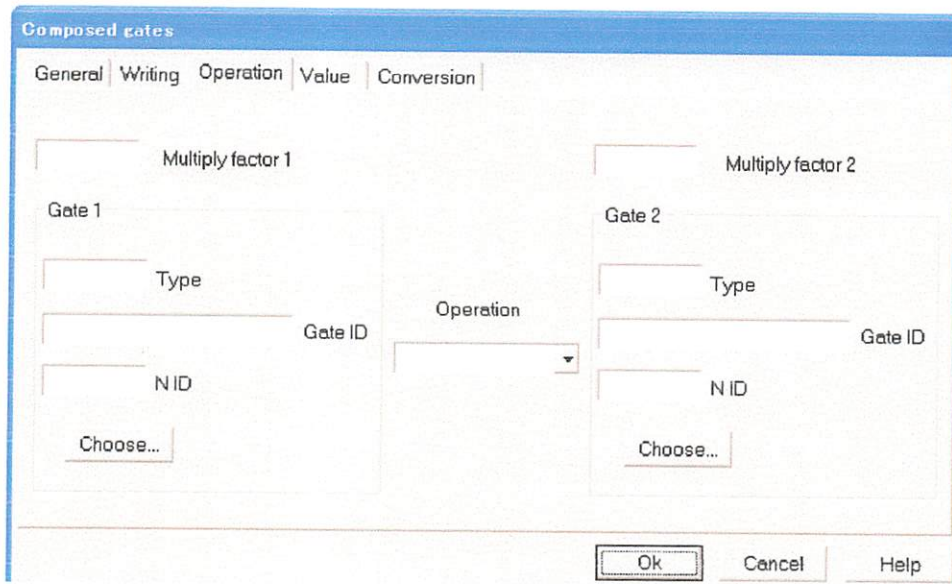
2.4.2.3.4 *Composed/Compound gate* ^[2]

Compound gate merupakan *gate* fiktif, yang tidak berhubungan dengan *register* di peralatan luar, tetapi merupakan hasil dari operasi matematik dari nilai satu atau dua *gate* lainnya.

Halaman setting terdiri dari :

- *General* : sama dengan *numerical gate*
- *Writing* : digunakan untuk setting *channel, device* dan alamat
- *Operation* (lihat gambar 2.39):
- *Gate 1 (Gate 2)* : merupakan *gate-gate* yang dioperasikan untuk menghasilkan suatu nilai dari *Compound gate*.
- *Multiply factor 1 (Multiply factor 2)* : nilai faktor perkalian dari nilai yang dibaca dari *gate 1 (gate 2)*.

- *Operation* : operasi yang dilakukan pada kedua gate. Operasi yang dapat dilakukan adalah :
 - + : penjumlahan
 - - : pengurangan
 - * : perkalian
 - / : pembagian
 - *Discard Hi* : Jika $\text{Gate 1} > \text{Gate 2}$ maka *compound gate* tidak dihitung, jika tidak *compound gate* = Gate 1
 - *Discard Lo* : Jika $\text{Gate 1} < \text{Gate 2}$ maka *compound gate* tidak dihitung, jika tidak *compound gate* = Gate 1
 - *Limit Hi* : Jika $\text{Gate 1} > \text{gate 2}$ maka *compound gate* = gate 2, jika tidak *compound gate* = Gate 1
 - *Limit Lo* : Jika $\text{Gate 1} < \text{Gate 2}$ maka *compound gate* = gate 2, jika tidak *compound gate* = gate 1
 - *Reset IF=* : Jika $\text{gate 1} = \text{gate 2}$ maka *compound gate* = 0, jika tidak *compound gate* = gate 1
- *Value*: sama dengan pada *numerical gate*
- *Conversion*: sama dengan pada *numerical gate*



Gambar 2.39
Parameter Operation pada Compound Gate. ^[2]

2.4.2.3.5 String Gate ^[2]

Terdiri dari :

- *General*: sama dengan digital gate
- *Sampling* : sama dengan *digital gate*
- *Value* : berisi *string* yang akan tampil pertama kali

2.4.2.3.6 Event/Alarm Gates ^[2]

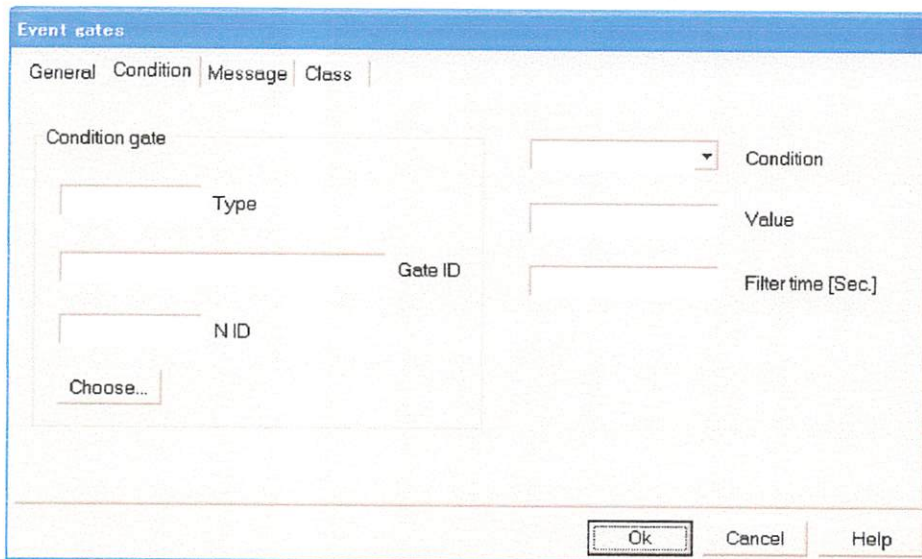
Terdapat empat halaman setting :

- *General* :

Terdiri dari

- Gate ID dan N ID : sama dengan numerical gate
- *Is Alarm* : jika dipilih, event diperlakukan sebagai alarm
- *Need acknowledge* : jika dipilih, event disimpan dalam tabel sampai dikonfirmasi oleh operator untuk mengaktifkannya.

- *Record on historical file* : sama dengan pada *numerical gate*
- *Access group* : sama dengan pada *numerical gate*
- Condition
 - Parameter condition terdiri dari (lihat gambar 2.40):
 - *Condition gate* : gate dimana akan dibaca nilainya untuk digunakan sebagai verifikasi suatu kondisi
 - *Condition* : kondisi yang harus dipenuhi supaya event aktif.
Kondisi yang dapat digunakan :
 - Value : nilai yang akan dibandingkan (tergantung dari kondisi yang dipilih)
- *Filter time* : waktu *minimum* yang harus dipenuhi oleh suatu kondisi untuk menghasilkan event.



Gambar 2.40
Parameter Condition pada Event Gate. [2]

- Message

- *Message* merupakan pesan yang akan ditampilkan (maksimum 80 karakter)
 - Untuk menambahkan informasi gate pada message digunakan tambahan %
 - %s : digunakan untuk menambahkan nilai *string* gate pada *message*
 - %lf : digunakan untuk menambahkan nilai numeric gate pada *message*
 - %x.ylf : x adalah jumlah digit yang akan ditampilkan, y adalah *optional* parameter menunjukkan jumlah decimal
- Class
- Class 1 : menyatakan nilai *numerik* yang digunakan untuk menyusun tabel dari event-event yang aktif (selama supervision).
 - Class 2 : menyatakan string *alpanumerik* yang digunakan untuk mengurutkan tabel dari event-event yang aktif (selama supervision).
 - Class 3-7 : menyatakan gate-gate dalam kelas yang akan ditrigger jika terjadi event

2.4.2.4. *Template Builder* ^[2]

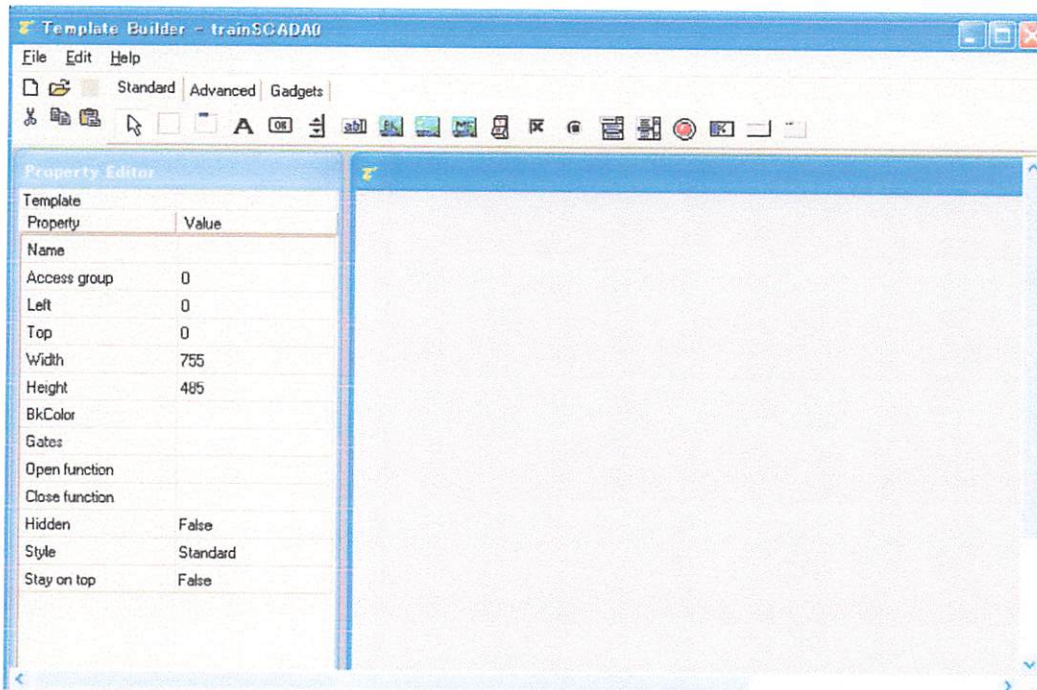
Template builder digunakan untuk merancang halaman supervisi dengan komponen-komponen antarmuka seperti button, label, status bar, dll.

2.4.2.4.1 *Template window* ^[2]

Template window menunjukkan halaman supervision (disebut *template*) yang akan ditampilkan/dijalankan saat Runtime seperti terlihat pada gambar 2.41.

Pada window ini ditempatkan semua komponen yang membentuk template tersebut dengan cara memilih komponen yang diinginkan dari *toolbar* dan mengkliknya pada template *window*. *Obyek* yang ditambahkan tersebut akan dipilih sehingga dapat dengan mudah diidentifikasi. Komponen dalam template dapat dipilih dengan menggunakan klik kanan mouse. Jika suatu komponen dipilih, Template builder akan menampilkan kotak kecil pada object *outline*, dan *property* editor akan menampilkan semua properties komponen tersebut.

Untuk memindahkan suatu objek klik pada objek tersebut dan pindahkan ke posisi yang baru, kemudian lepaskan tombol kiri mouse.



Gambar 2.41
Template window. [2]

2.4.2.4.2 Deskripsi Komponen

2.4.2.4.2.1 BkBitmap

BkBitmap (*background* bitmap) menyatakan image yang dapat ditambahkan pada template untuk menampilkan gambar yang statik (tidak berubah selama proses supervisi).

Untuk menentukan gambar yang ditampilkan, tekan button pada property “image”, selanjutnya akan muncul menu yang berisi direktori gambar. Pilih gambar bitmap dari direktori yang ada, maka *preview* dari gambar akan ditampilkan. Klik OK untuk mengkonfirmasi gambar yang dipilih.

Selain mengambil gambar dari direktori Winlog, dua buah tombol di sebelah kanan (MsPaint dan *Symbol Factory*) dapat digunakan untuk mengambil gambar dari luar.

Untuk mengimpor gambar dari luar, *copy* gambar ke *clipboard* (gunakan perintah Edit|*Select All* lalu Edit|*Copy* pada MSPaint, atau perintah *Copy* dari *Symbol factory*), selanjutnya tekan Paste.

2.4.2.4.2.2 Bitmap

Tidak seperti BkBitmap yang menampilkan gambar statik, Bitmap dapat digunakan untuk menampilkan gambar yang berubah selama supervisi, sehingga gambar berubah sesuai dengan kondisi *plant* yang dimonitor.

Bitmap dapat diisi dengan suatu daftar gambar dengan dinyatakan kondisi dari setiap gambarnya. Jika suatu kondisi terpenuhi maka gambar yang bersangkutan yang akan ditampilkan.

Beberapa *property* dari Bitmap ini adalah :

- **Bitmap states:** menyatakan gambar dan kondisi dari bitmap. Untuk memilih kondisi tekan button pada *property*, dan sebuah *window* akan muncul. Di setiap baris di tabel window ini ditampilkan gambar yang akan ditampilkan beserta kondisi yang harus dipenuhi. Untuk menambah status baru, tekan tombol *Add*. Tombol Edit digunakan untuk mengedit gambar dan kondisi yang sudah ditambahkan. Tombol *Remove* digunakan untuk menghapus status dari daftar. Untuk mengurutkan gambar bitmap dapat digunakan tombol dengan simbol anak panah.

Untuk mengedit kondisi digunakan tombol Edit *conditions* yang menampilkan daftar kondisi dari setiap status yang dipilih.

Tekan tombol *Add* untuk menambah kondisi, atau Edit untuk mengedit/memodifikasi kondisi.

Untuk mengisi suatu kondisi dilakukan sbb:

- o Pilih gate yang digunakan sebagai penentu kondisi
- o Pilih tipe kondisi dan nilai yang digunakan sebagai pembanding

- ***Color replacement***

Digunakan untuk mengubah warna Bitmap, yang dapat berupa :

- o *Transparent color* : dengan warna transparan, maka latar belakang di belakang Bitmap akan tampak.
- o *Replace color with...*: digunakan untuk mengganti warna Bitmap.

- ***On Click***

Menyatakan operasi yang dilakukan ketika tombol ditekan. Daftar operasi yang dapat dilakukan adalah:

- *Call function* : memanggil suatu fungsi
- *Stop function* : menghentikan suatu fungsi
- *Open template* : membuka suatu template
- *Close template* : menutup suatu template
- *Apply changes* : melakukan perubahan komponen
- *Undo changes* : membatalkan perubahan komponen

- *X Animation*

Menyatakan gate yang digunakan untuk membaca koordinat-X dari Bitmap. Dengan property ini, posisi Bitmap dapat dipindahkan ke posisi yang dinyatakan oleh gate. Nilai 0 menyatakan tepi kiri dari template.

- *Y Animation*

Menyatakan gate yang digunakan untuk membaca koordinat-Y dari Bitmap. Dengan property ini, posisi Bitmap dapat dipindahkan ke posisi yang dinyatakan oleh gate. Nilai 0 menyatakan tepi atas dari template

2.4.2.4.2.3 *Button*

Fungsi utama dari *button* adalah untuk melakukan berbagai macam operasi (selama supervisi) ketika button ditekan.

Beberapa *property* dari Bitmap adalah:

- *Cursor* : menyatakan bentuk kursor mouse ketika menyorot button selama supervisi. Terdapat enam pilihan bentuk kursor : *White, Red, Yellow, Click, Double Click*.

- Label : menyatakan teks yang ditampilkan pada Button
- On *Click* : menyatakan operasi yang dilakukan ketika tombol ditekan. Daftar operasi yang dapat dilakukan sama seperti pada On *Click* Bitmap.
- Enable : object akan enable (berfungsi) jika satu dari kondisi yang diberikan terpenuhi. Objek akan selalu *enable* jika tidak ada kondisi yang diberikan.

2.4.2.4.2.4 Edit

Edit digunakan untuk menyatakan suatu nilai yang diberikan pada gate.

Beberapa *property* pada Edit:

- Gate : menyatakan gate dimana nilai yang ditulis pada Edit akan dituliskan.
- *Need apply* : menyatakan kapan nilai yang diubah akan dituliskan ke gate.

Terdapat dua pilihan:

- o No : nilai akan dituliskan ke gate ketika menekan *Return (Enter)* atau Tab.
- o Yes : nilai akan dituliskan ke gate ketika user memberikan konfirmasi dari luar, misalnya dengan menggunakan *Button* dengan *property On Click* *diset Apply*
- *Validation string* : menyatakan *string* yang digunakan sebagai skema untuk penulisan di Edit.

Simbol yang digunakan untuk validasi adalah :

- o # : hanya menerima digit
- o ? : hanya menerima huruf (huruf besar atau kecil)

- **&**: hanya menerima huruf dan mengubahnya ke huruf besar
- **@**: menerima semua karakter (huruf, digit)
- **|** : menerima semua karakter (huruf, digit) dan mengubahnya ke huruf besar
- **[]**: karakter optional

Contoh : jika diinginkan mengisi Edit dengan 3 angka, *sign* atau *unsigned*, dengan paling besar dua desimal di belakang koma, maka digunakan validation string : “[+][-]###.[#][#]”.

- **Enable** : *object* akan *enable* (berfungsi) jika satu dari kondisi yang diberikan terpenuhi. Objek akan selalu *enable* jika tidak ada kondisi yang diberikan.

2.4.2.4.2.5 Gauge

Gauge merupakan komponen yang digunakan untuk menampilkan nilai gate secara grafik.

Beberapa property dari Gauge:

- **Gate** : menyatakan gate dimana nilainya akan dibaca dan ditampilkan selama *supervisi*. Hanya *numerical* atau *compound* gate yang dapat digunakan.
- **Minimum value**: nilai minimum Gauge. Jika Gauge membaca nilai minimum ini dari suatu gate, maka bar dengan ukuran-0 (tidak ada bar) akan ditampilkan.

- *Maximum value*: nilai maksimum Gauge. Jika Gauge membaca nilai maksimum ini dari suatu gate, maka akan ditampilkan bar dengan ukuran sama dengan lebar (atau tinggi) Gauge.
- *Label* :menyatakan teks yang akan ditampilkan. Jika teks yang ingin ditampilkan menyatakan nilai yang dibaca dari Gate, maka digunakan “%g” untuk bilangan desiman dan “%d” untuk integer. Contoh: Jika ingin menampilkan suhu yang dibaca dari gate, label dapat diset “%g oC”.
- *Segments*: menyatakan jumlah segmen yang membentuk bar dari *Gauge*. Perubahan nilai segment tidak ditampilkan pada *Template Buildr*, tetapi hanya ditampilkan pada saat supervisi
- *Space between segments*: menyatakan jumlah piksel yang memisahkan satu segmen dengan segmen lainnya.
- *Direction* : menyatakan arah gauge. Terdapat dua pilihan arah : horisontal dan vertikal.

2.4.2.4.2.6 Label

Label merupakan teks yang dapat digunakan menampilkan informasi berupa teks.

Label dapat digunakan sebagai label biasa, atau untuk menuliskan nilai yang dibaca dari suatu gate.

Beberapa *property* dari Label:

- *Gate* : menyatakan gate dimana nilainya dibaca dan dituliskan ke label.
- *Label* : menyatakan teks yang akan ditampilkan pada label. Label ini mempunyai format seperti label pada *Gauge* untuk menampilkan format tulisan.

2.4.2.4.2.7 Led

Led digunakan untuk menampilkan status gate secara grafik berbentuk Led.

Beberapa property dari Led:

- *OFF appearance*: menyatakan tampilan Led pada saat kondisi OFF. Gambar pada LED dapat dipilih dari gambar-gambar *default* yang ada, atau dari suatu file gambar.
- *ON appearance* : menyatakan tampilan Led pada saat kondisi ON. Gambar pada LED dapat dipilih dari gambar-gambar *default* yang ada, atau dari suatu file gambar.
- *Led ON conditions* : menyatakan daftar kondisi supaya Led ON. Properti ini hampir sama dengan *property State Condition* pada komponen Bitmap.
- *Flashing* :

2.4.2.4.2.8 Metafile

Sama dengan komponen Bitmap yang dapat menampilkan gambar yang berubah-ubah selama *supervisi*. Perbedaan dengan Bitmap adalah format gambar yang dipilih adalah Metafile.

2.4.2.4.2.9 Switch

Switch merupakan komponen yang berfungsi sebagai *switch/saklar* yang mempunyai status ON atau OFF. Selama *supervisi*, user dapat mengubah status *switch* dengan cara meng-klik komponen *Switch*. *Switch* dapat diprogram untuk bereaksi sesuai status yang dibaca dari gate. Pada saat Runtime, nilai gate yang

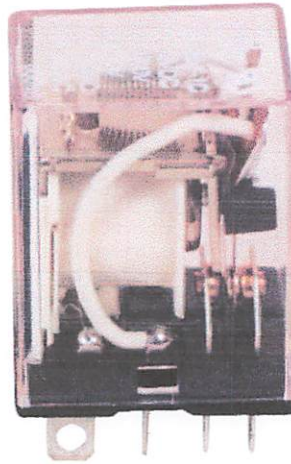
bersangkutan dicek, jika nilainya memenuhi kondisi ON, maka Switch akan berubah pada status ON, dan sebaliknya.

Beberapa *property* dari *Switch*:

- *Image* : menyatakan tampilan dari switch yang dapat dipilih dari gambar *default* yang ada, atau dari file gambar.
- *ON condition* : menyatakan gate yang akan dituliskan status dari *switch*. Terdapat dua kemungkinan penulisan status ke gate: bit atau *mask writing*.
 - o *Bit writing* : Ketika *Switch* ON, pada bit yang ditentukan (0-31) akan dituliskan nilai ON (0 atau 1), dan sebaliknya.
 - o *Mask writing* : bit dari gate yang dituliskan hanya yang mempunyai *mask* yang sama dengan 1, yaitu jika *Switch* ON, maka bit dari gate yang bernilai 1 akan dituliskan nilai ON.

2.5 Relay ^[3]

Relay merupakan salah satu jenis saklar *magnetic* yang dapat memutuskan atau menghubungkan kontak – kontak dengan arus yang dialirkan ke kumparan (inti). Sebuah *relay* terdiri dari kumparan dan inti, yang mana bila dialiri arus kumparan tersebut akan menjadi magnet dan menutup atau membuka kontak. Keuntungan *relay* adalah dapat menghubungkan daya yang besar dengan member daya yang kecil pada kumparannya.



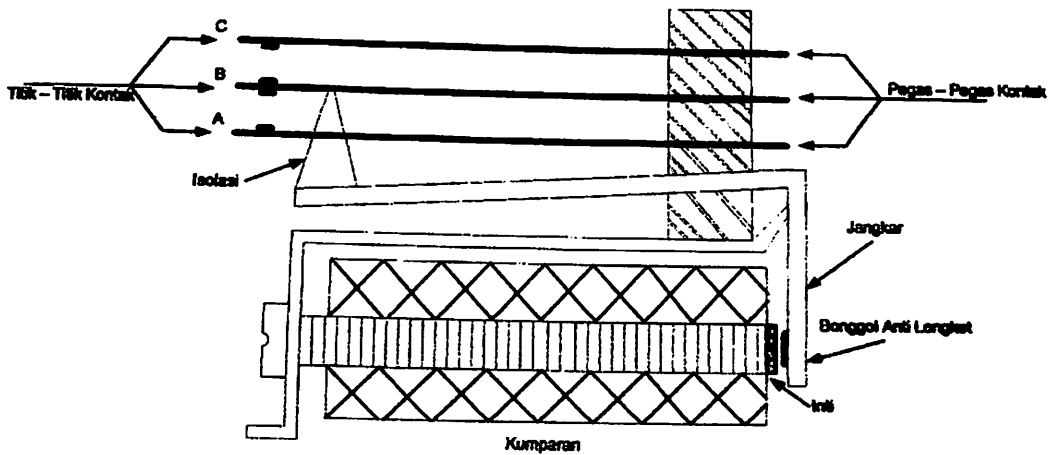
Gambar 2.42
Bentuk Fisik Relay ^[3]

Pada dasarnya prinsip kerja *relay* sama dengan kontaktor, yang mana berfungsi untuk membuka dan menutup kontak listrik yang dikontrol dengan prinsip kerja elektromagnetik. Kerja dari *relay* tersebut apabila arus mengalir didalam kumparan yang memiliki inti besi lunak akan tertarik dan bergerak menggelinding pada engsel (*pivot*).

Relay dapat menggulingkan kalau daya gaya magnet dapat mengarahkan gaya pegas yang mengalahkannya, maka kontak pun menutup. Besarnya gaya magnet ditentukan oleh kuat medan magnet pada celah udara antara jangkar dan inti besi, sedangkan kuat medan magnet tergantung pada jumlah lilitan kumparan dan kuat arus, kuat medan magnet ditetapkan juga oleh besar resistansi magnet dalam sirkuit kemagnetan. Kuat medan dicelah udara akan semakin kuat bila letak jangkar semakin dekat dengan inti. Jarak jangkar dan inti dapat diatur dengan menyetel pencairan pegas.

Seperti halnya kontaktor, *Relay* dapat menggerakkan beberapa kontak sekaligus hanya dengan suatu kumparan jangkar.

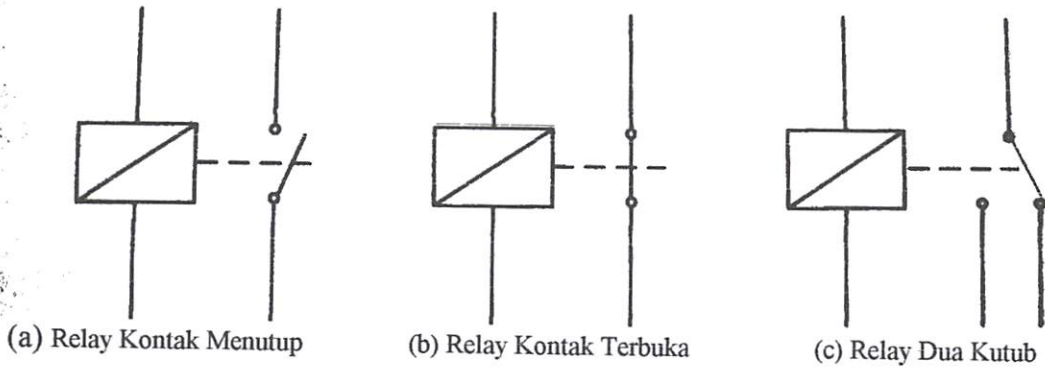
Ada dua jenis Relay, yaitu : (1) Relay yang bekerja dengan arus bolak – balik, dan (2) Relay yang bekerja dengan arus searah. Jenis Relay yang bekerja dengan arus bo;ak – balik tidak bisa bekerja pada alat – alat elektronik.



Gambar 2.43
konstruksi relay jenis kontak tukar [3]

Pada gambar diatas bila kumparan dialiri arus listrik, maka akan timbul medan magnet dan menarik jangkar, sehingga kontak antara A dan B putus (membuka), kontak B dan C menutup. Jenis Relay ini dinamakan dengan kontak tukar.

Jenis lainnya adalah jenis Relay dengan kontak menutup, dimana apabila diberi arus listrik, maka kontak – kontaknya menutup. Relay dengan kontak membuka dimana apabila kumparan Relay diberi arus listrik maka kontak – kontaknya akan membuka. Terdapat juga jenis Relay dengan dua kutub (Bi – Polar) dimana Relay ini mempunyai dua kumparan dan dua kondisi kerja. Bila Relay tidak diberi arus listrik, maka kontak Bi bebas, tidak menghubungkan kemana – mana, kalau kumparan 1 terhubung dengan arus listrik, maka kontak B menghubungkan kontak A. Kalau kumparan 2 terhubung dengan arus listrik, maka kontak B terhubung dengan kontak C.



Gambar 2.44
jenis – jenis relay ^[3]

2.6 MCB ^[3]

Fungsi MCB adalah untuk pengaman terhadap beban lebih atau arus hubung singkat. MCB akan bekerja memutuskan rangkaian dari sumber. Pengaman ini memutuskan secara otomatis kalau arusnya melebihi rating arus nominal yang dimiliki oleh MCB.



Gambar 2.45
Bentuk Fisik MCB (*Magnetic Circuit Breaker*)

Pada prinsipnya pengaman ini memberikan pengaman termos maupun Relay Elektronik. Pengaman termos digunakan untuk melindungi beban lebih. Jika arus yang melewati MCB lebih besar dari arus nominal MCB maka arus akan terputus. Pemutus secara termos berlangsung dengan kelambatan, dimana

lamanya waktu pemutusan tergantung besar arusnya, sedangkan pengaman elektronik digunakan sebagai pelindung apabila terjadi hubungan singkat.

Bedasarkan waktu pemutusan, pengamanotomatisasi ini dibagi atas :

1. MCB Type L (Untuk Hantaran)

Pada tipe ini pengaman thermisnya disesuaikan dengan meningkatnya suhu hantaran. Kalau terjadi beban lebih dan suhu hantarannya melebihi batas tertentu maka elemen dwi logamnya akan memutuskan arusnya.

Kalau terjadi hubung singkat, arusnya diputuskan oleh pengaman elektromagnetiknya. Untuk arus bolak – balik yang sama dengan 4 ampere sampai 6 ampere, pemutusan arusnya berlangsung dalam waktu 0.2 detik.

2. MCB Type H (Untuk Instalasi Rumah)

Pengaman thermos jenis sama dengan MCB tipe L tetapi pengaman elektromagnetiknya memutuskan dalam waktu 0,2 detik, jika arusnya sama dengan 2,5 ampere untuk arus bolak – balik atau sama dengan 4 ampere untuk arus yang rendah pun harus diputuskan dengan cepat. Jadi kalau terjadi gangguan tanah, bagian – bagian yang tersebut dari logam tidak akan bertegangan.

3. MCB Type G (Untuk Motor – Motor Listrik)

MCB jenis ini digunakan untuk pengaman motor – motor listrik kecil, untuk arus bolak – balik atau searah, alat – alat listrik dan juga rangkaian besar untuk penerangan, misalnya bengkel atau pabrik. Pengaman elektromagnetiknya berfungsi pada 8 ampere sampai 11 ampere untuk arus bolak – balik atau 14 ampere untuk arus searah.

2.7 Buzzer

Buzzer akan aktif jika sensor suhu atau inkubator tidak bekerja dengan sesuai settingannya dan akan mengeluarkan bunyi sebagai tanda apabila inkubator tidak bekerja. Gambar fisik dari buzzer adalah sebagai berikut :



Gambar 2.46 Bentuk Fisik Buzzer

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1 Pendahuluan

Dalam bab ini akan membahas mengenai perancangan dan pembuatan keseluruhan sistem perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan dalam Sistem Pengendalian Multi Inkubator Menggunakan PLC *Smart Relay Zelio Logic* Berbasis SCADA.

3.2 Perancangan Perangkat Keras (*Hardware*)

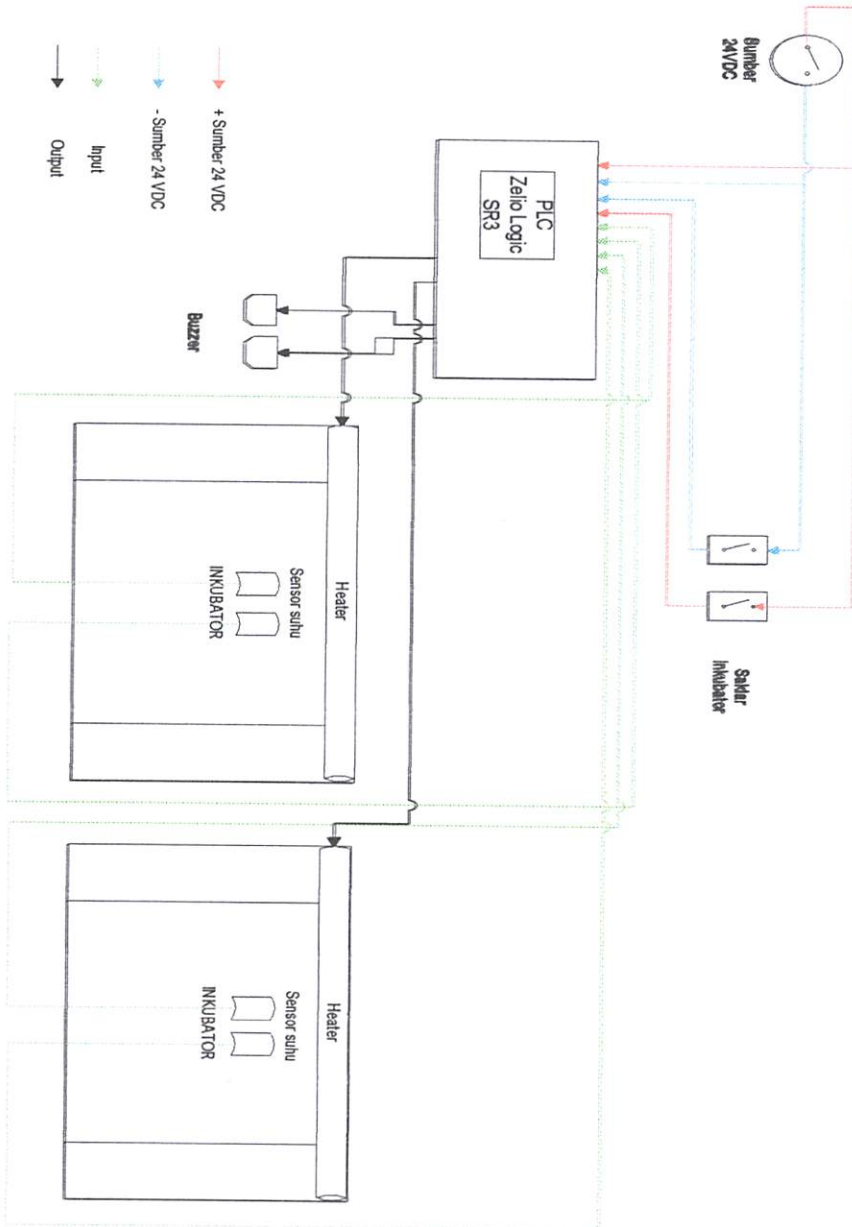
Pembuatan Sistem Pengendalian Multi Inkubator ini dibuat dalam bentuk aslinya dengan perancangan dan prinsip kerja yang sesuai dengan inkubator yang sebenarnya.

Sistem pengendalian yang diterapkan dalam perancangan alat ini yaitu menggunakan PLC *Smart Relay Zelio Logic* yang digunakan untuk mengendalikan inkubator lebih dari satu secara otomatisasi baik itu sensor suhu, heater, fan, dan juga menggunakan *modbus* untuk SCADA (software winlog) yang digunakan untuk memonitor dan mengontrol inkubator dari jarak jauh menggunakan software melalui PC / Laptop, dan tanpa harus mendatangi tempat dimana inkubator tersebut berada.

Komponen utama dari sistem ini adalah PLC *Smart Relay Zelio Logic* yang berfungsi sebagai unit pengendali inkubator lebih dari satu secara otomatisasi, *modbus* berfungsi sebagai komunikasi antara PLC *Smart Relay Zelio Logic* PC/Laptop, SCADA (software winlog) yang berfungsi mengatur dan

mengontrol inkubator dari jarak jauh dan juga *heater* yang berfungsi sebagai penghangat ruangan dalam inkubator, serta *fan* yang berfungsi untuk mengeluarkan udara panas dari inkubator atau untuk sirkulasi udara.

3.2.1 Blok Diagram Sistem



Gambar 3.1
Blok Diagram Sistem Pengendalian Multi Inkubator Menggunakan PLC *Smart Relay Zelio Logic* Berbasis SCADA.

Penjelasan tiap block diagram

- *Smart Relay Zelio Logic*

Digunakan sebagai pengelolah data dari masukan input hingga keluaran output

- Sensor Suhu (IC LM 35)

Digunakan sebagai pendeteksi suhu didalam inkubator

- *Heater*

Digunakan untuk memberikan udara panas kedalam inkubator

- *Fan*

Digunakan untuk mengeluarkan udara panas

- Box Inkubator

Digunakan sebagai simulasi otomatisasi inkubator

- Buzzer

Digunakan untuk indikator bunyi

- Saklar On/Off

Digunakan untuk memfungsikan inkubator 1 dan inkubator 2.

Prinsip Kerja Sistem :

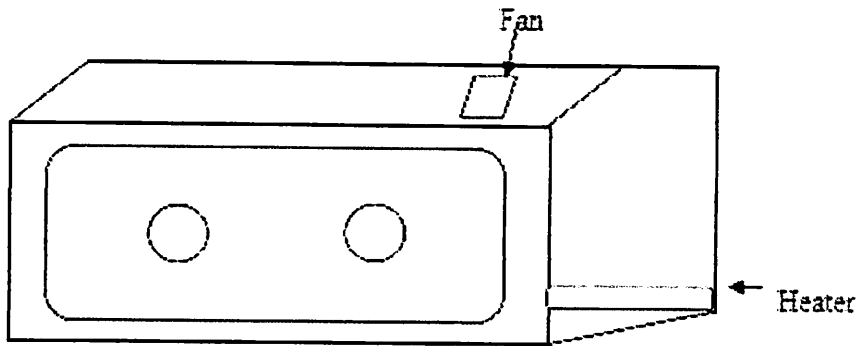
1. Pada setiap inkubator dipasang dengan 2 buah sensor suhu (IC LM 35) sebagai pendeteksi suhu didalam inkubator, yang terletak pada sisi belakang inkubator dan diletakkan berdekatan ditengah inkubator agar sensor suhu dapat mendeteksi suhu dalam inkubator secara merata. Dalam inkubator juga terdapat *heater* dan juga *fan* yang digunakan untuk mengatur suhu dalam inkubator dan sirkulasi udara.

2. Sistem dalam kondisi On apabila tombol pada modul PLC dikondisikan Running. Ketika sistem On, *Smart Relay Zelio Logic* SR3 tidak akan membaca sensor suhu (IC LM 35) pada inkubator dikarenakan saklar pada box panel harus dikondisikan On terlebih dahulu. Sistem On dan Saklar On *Smart Relay Zelio Logic* akan membaca sensor suhu apakah suhu dalam inkubator sudah mencapai 36°C - 37°C dan inkubator pun bekerja.
3. Pada saat sensor suhu (IC LM 35) mendeteksi suhu kurang dari suhu 36°C - 37°C , sistem akan menghidupkan *heater* (elemen panas) hingga suhu mencapai 37°C . Pada saat sensor suhu (IC LM 35) sudah mendeteksi suhu mencapai 37°C , maka secara otomatis *heater* (elemen pemanas) akan mati. Heater (elemen pemanas) mati maka suhu akan turun dan jika sensor suhu sudah mendeteksi suhu 36°C lagi, maka *heater* (elemen panas) akan hidup kembali, sedangkan *fan* (kipas) akan hidup terus menerus selama sistem On sebagai sirkulasi udara didalam inkubator.
4. Pada saat sensor suhu (IC LM 35) setelah bekerja mendeteksi suhu didalam inkubator kemudian tiba-tiba tidak bekerja/tidak mendeteksi suhu, atau dianggap sensor suhu rusak baik sensor primer maupun back up, maka buzzer akan bekerja atau berbunyi untuk memberikan tanda bahwa ada masalah pada inkubator tersebut.
5. SCADA atau Software Winlog Lite 2.06.74 sendiri akan menampilkan pada saat inkubator bekerja ke dalam PC/Laptop, baik berupa suhu yang ada didalam inkubator saat sistem On, setingan suhu yang diperlukan inkubator dapat diseting melalui PC/Laptop serta juga dapat memonitor

kerja dari *heater* (elemen panas) dan juga dapat memberi tahu apabila terjadi kerusakan pada sensor suhu baik primer maupun back up.

3.2.2 Perancangan Inkubator

Perancangan inkubator bertujuan untuk menentukan letak pemasangan dari *heater* dan *fan* berdasarkan fungsinya masing – masing. Perancangan pemasangan *heater* bertujuan untuk menghasilkan udara panas yang merata pada inkubator dan *fan* digunakan untuk mengeluarkan udara panas jika udara panas didalam inkubator berlebihan atau sebagai sirkulasi udara didalam inkubator.

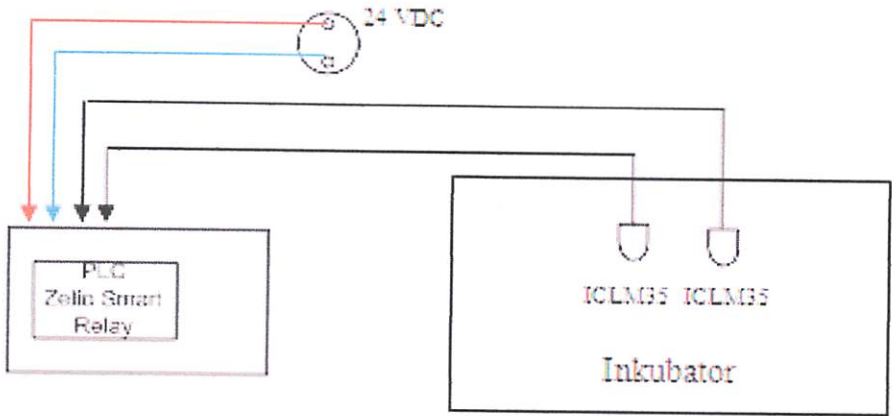


Gambar 3.2
Bentuk Fisik Inkubator

3.2.3 Perancangan Pemasangan Sensor Suhu (IC LM 35)

Perancangan pemasangan sensor suhu (IC LM 35) ini bertujuan untuk menentukan tempat pemasangan berdasarkan fungsinya dengan meletakkan sensor suhu (IC LM 35) pada sisi belakan inkubator dan berdekatan ditengah inkubator agar dapat mendeteksi suhu dalam inkubator secara merata, dan juga agar *Smart Relay Zelio Logic SR 3* dapat membaca fungsinya dari sensor tersebut.

Sensor suhu IC LM 35 diletakkan pada sisi belakang inkubator dan berdekatan ditengah, dimana agar suhu didalam inkubator dapat dideteksi secara merata antara sisi kanan dan sisi kiri.

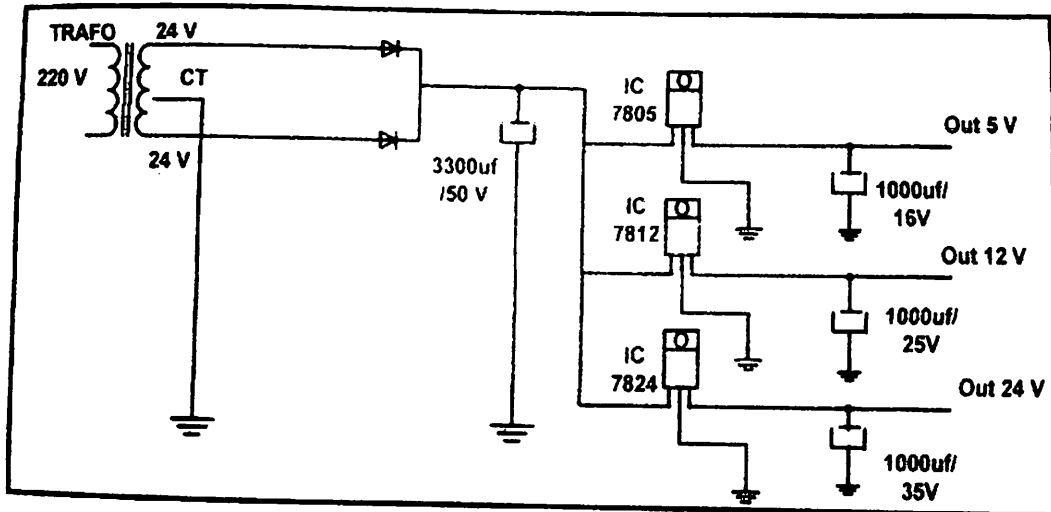


Gambar 3.3
Perancangan letak pemasangan sensor suhu IC LM 35 pada setiap Inkubator

3.2.4 Power Supply

Untuk suplai tegangan pada sistem, menggunakan tegangan PLN220 VAC, *Power Supply* 24 VDC yang digunakan untuk mensuplai *Smart Relay Zelio Logic SR 3*, dan *Power Supply* 5 – 12 VDC digunakan untuk mensuplai sensor suhu IC LM 35.

Dibawah ini adalah gambar rangkaian *power supply* :



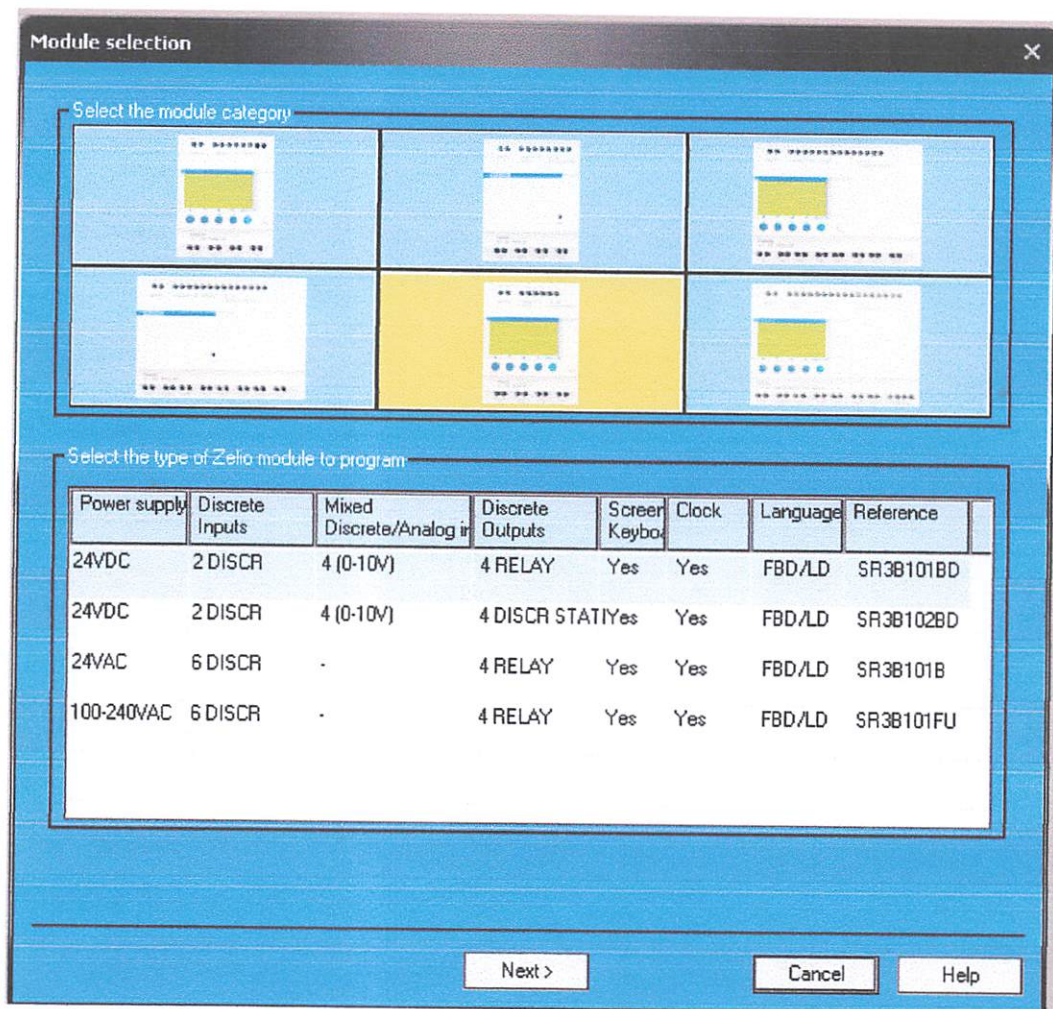
Gambar 3.4
Power Supply Tegangan Untuk Sistem

3.3 Perancangan Perangkat Lunak (Software)

3.3.1 Perancangan Software Zelio Soft 2

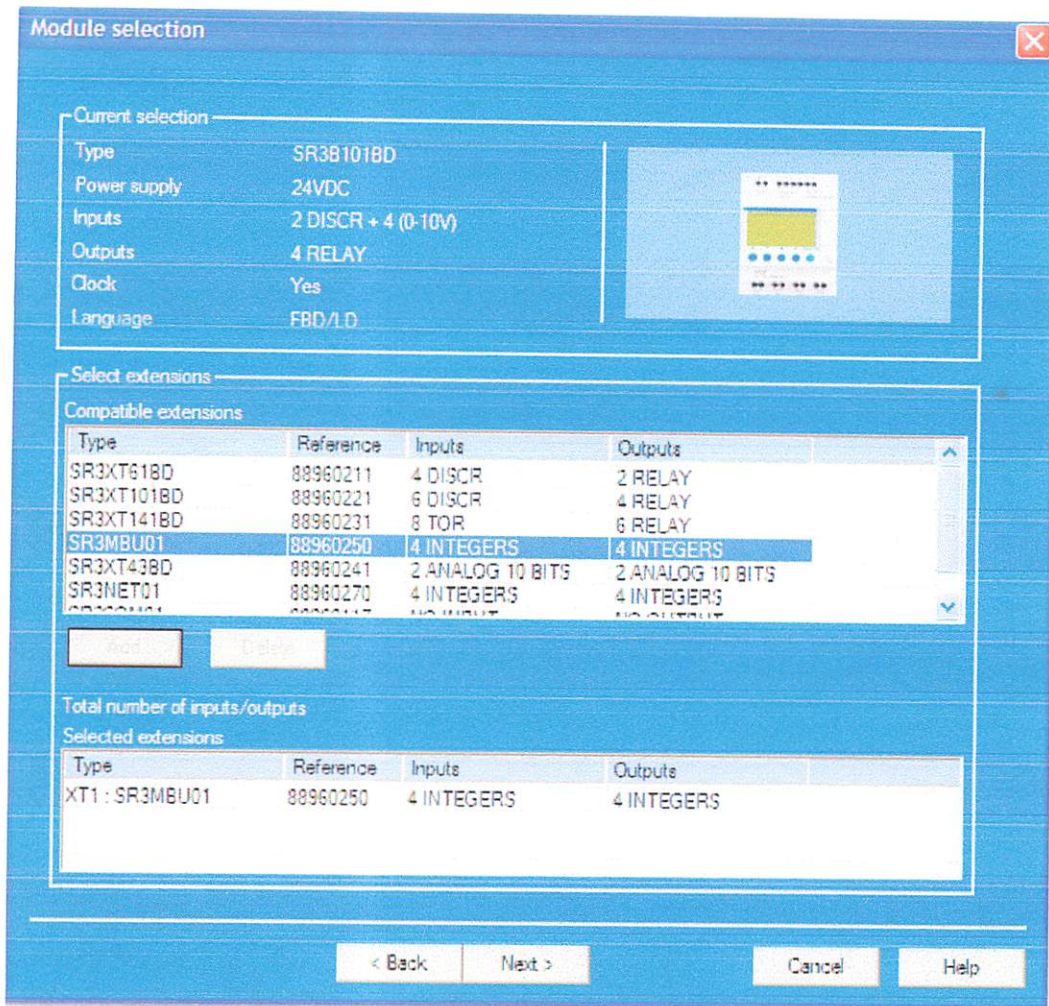
Perancangan perangkat lunak (*software*) pada Perancangan Dan Pembuatan Sistem Pengendalian Multi Inkubator Menggunakan PLC *Smart Relay Zelio Logic* Berbasis SCADA yaitu menggunakan software Zelio Soft 2 dan juga menggunakan software winlog untuk komunikasi SCADA nya.

Sebelum menggunakan software *Zelio Soft 2* terlebih dahulu melakukan pemrograman, pertama terlebih dahulu memilih konfigurasi dari tipe *Smart Relay* yang digunakan pada sistem seperti yang ditunjukkan pada gambar berikut :



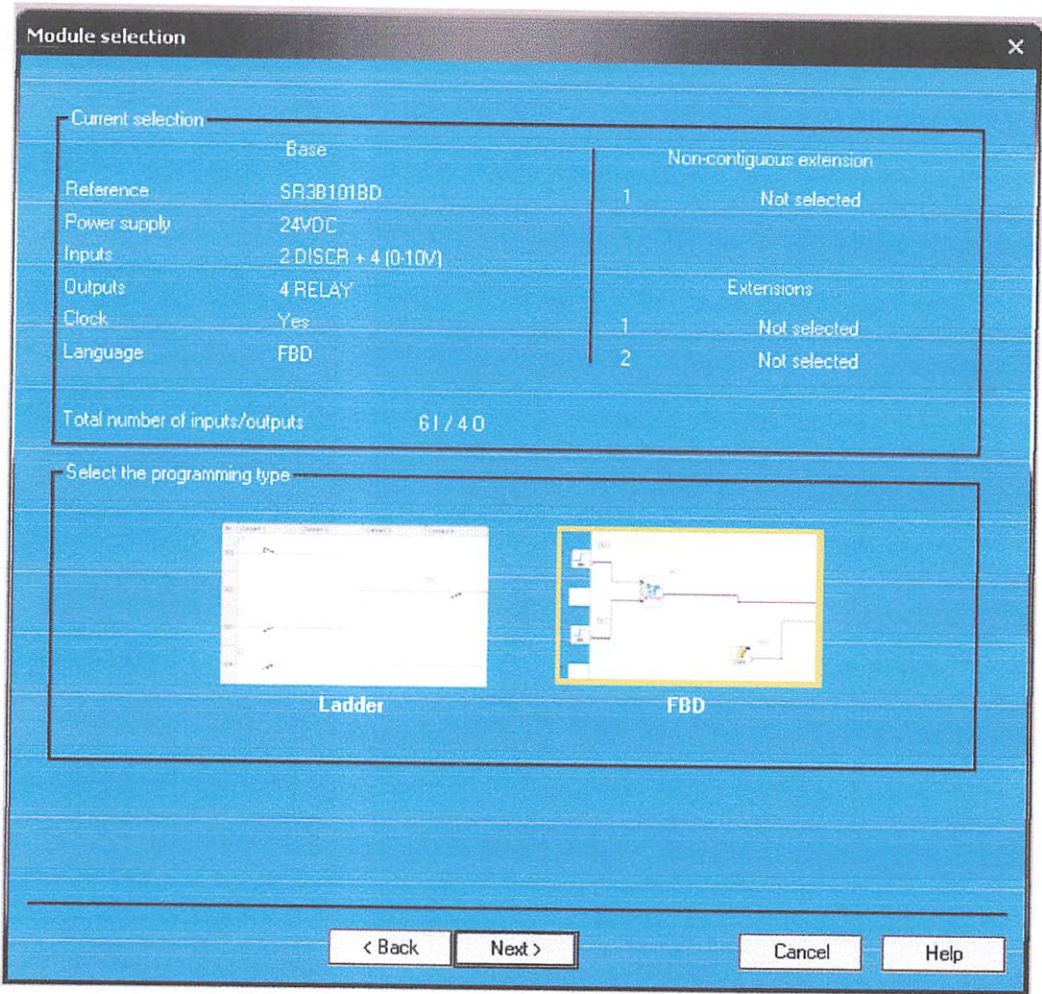
Gambar 3.5
Konfigurasi Smart Relay

Konfigurasi dicocokkan dengan *Smart Relay* yang digunakan seperti : suplai tegangan, jumlah I/O yang digunakan serta tipe dari *Smart Relay* itu sendiri. Selain itu setelah mencocokkan tipe *Smart Relay* yang digunakan, dapat juga menambahkan modul tambahan seperti *modul compatible* atau *modbus* sesuai dengan kebutuhan yang diperlukan oleh sistem yang akan dibuat seperti pada gambar berikut :



Gambar 3.6
Konfigurasi Modul Tambahan

Setelah mengkonfigurasi modul tambahan, langkah selanjutnya mamilih bahasa pemrograman yang digunakan. *Software Zelio Soft 2* memiliki 2 bahasa pemrograman yaitu *Ladder* dan bahasa *Function Block Diagram (FBD)* seperti pada gambar berikut :

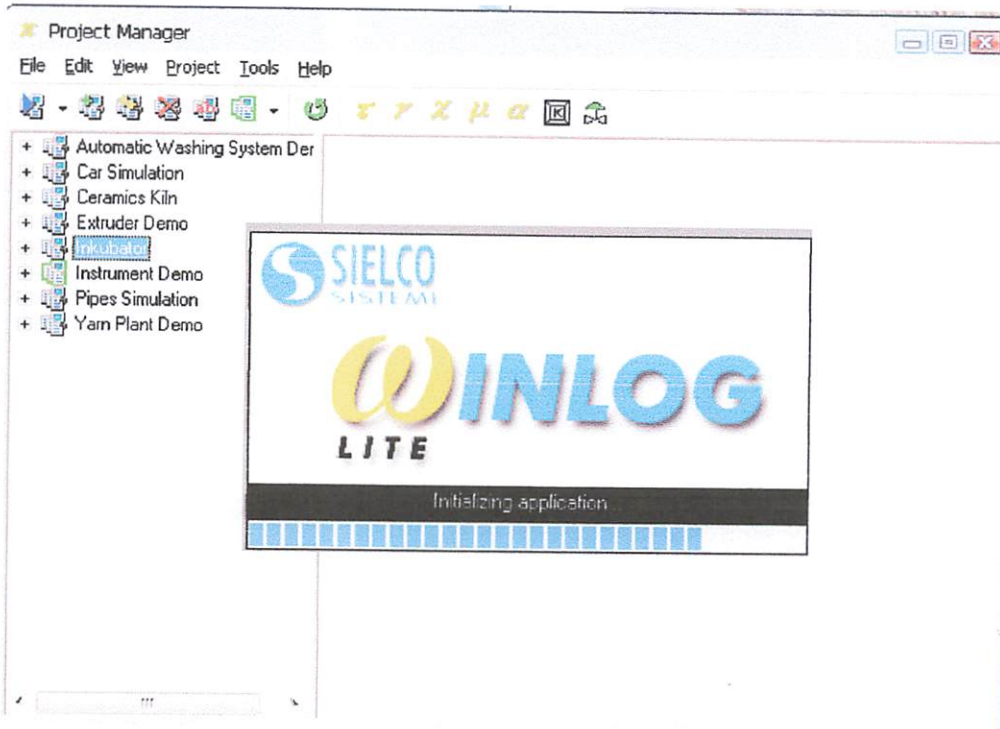


Gambar 3.7
Konfigurasi Ladder dan Function Block Diagram (FBD)

Bahasa pemrograman yang digunakan dalam pembuatan sistem ini adalah bahasa *Function Block Diagram* (FBD). Sebelum program ditransfer kedalam Smart Relay, terlebih dahulu dapat disimulasikan dalam *Software Zelio Soft 2* sehingga apabila terjadi kesalahan pemrograman dapat diketahui sebelum program ditransfer kedalam *modul Smart Relay*.

3.3.2 Perancangan Software Winlog Lite 2.06.74

Perancangan perangkat lunak (*software*) pada Perancangan Dan Pembuatan Sistem Pengendalian Multi Inkubator Menggunakan PLC *Smart Relay Zelio Logic* Berbasis SCADA yaitu juga menggunakan software Winlog Lite 2.06.74.



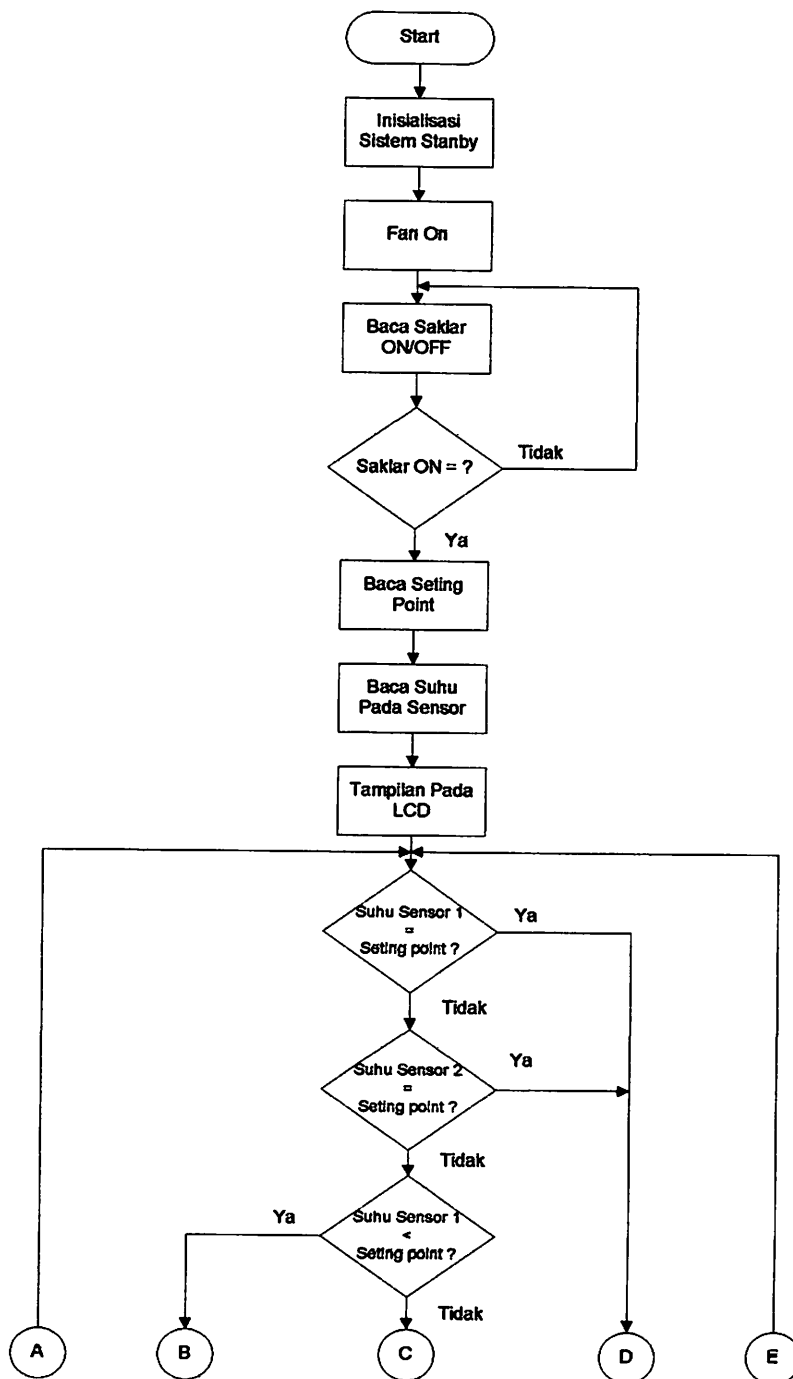
Gambar 3.8
Software Winlog Lite 2.06.74

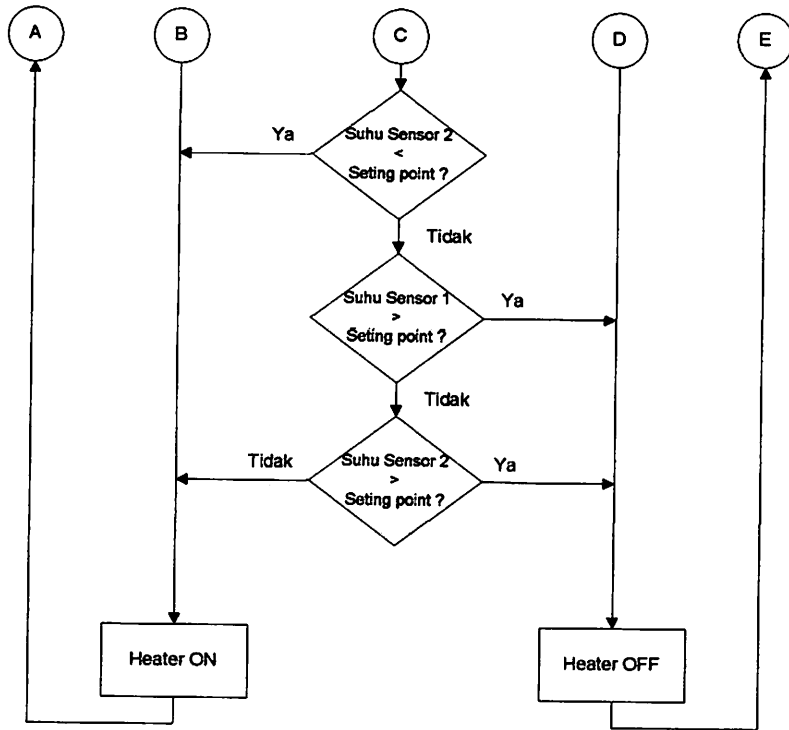
Bahasa pemrograman yang digunakan Software Winlog Lite 2.06.74 yaitu bahasa campuran antara bahasa C dan Delphi, bahasa C dan *Delphi* disini digunakan untuk tambahan guna mendukung software winlog agar dapat berkomunikasi secara kompleks. Sebelum program dihubungkan dengan *Smart relay / Modbus*, terlebih dahulu dapat disimulasikan dalam software Winlog Lite 2.06.74 itu sendiri, sehingga apabila terjadi kesalahan pemrograman atau

kekurang sempurnaan hasil yang didapatkan, dapat diketahui sebelum program dihubungkan dengan *modul Smart Relay / Modbus*.

3.3.3 Diagram Alir Perancangan

Flowchart rangkaian secara keseluruhan adalah sebagai berikut :





Gambar 3.9
Flowchart Sistem Inkubator 1 dan Inkubator 2

3.4 Spesifikasi Alat

Pada Perancangan Dan Pembuatan Sistem Pengendalian Multi Inkubator Menggunakan PLC *Smart Relay Zelio Logic* Berbasis SCADA ini, mempunyai spesifikasi alat sebagai berikut :

1. Nama Alat :

Perancangan Dan Pembuatan Sistem Pengendalian Multi Inkubator Menggunakan PLC *Smart Relay Zelio Logic* SR3PACKBD dan SR3MBU01BD (Zelio Modul Komunikasi) serta kabel D-485/9D RS232 to RS485 *Converter*.

2. Fungsi Alat :

Mengontrol secara otomatis *heater* dan *fan* disetiap inkubator agar dapat mencapai suhu yang telah ditentukan dalam inkubator tersebut, serta

berkomunikasi dengan PC/Laptop untuk menginformasikan dan juga mengontrol kerja dari inkubator melalui PC/Laptop secara bersamaan.

3. Keistimewaan atau kemampuan tambahan alat adalah :

- Menggantikan sistem kontrol yang manual menjadi otomatis pada inkubator.
- Menggunakan sensor suhu IC LM 35 yang difungsikan sebagai pendeteksi suhu didalam inkubator yang memiliki akurasi 0.5°C dalam suhu ruangan dengan konfigurasi $10\text{ mV}/^{\circ}\text{C}$.
- Sistem bisa dimonitoring manual menggunakan *software Zelio Soft 2*, untuk kontrol manual dan mengetahui semua sistem bekerja dengan baik.
- Sistem dapat memberikan informasi dan juga dapat dikontrol melalui PC/Laptop dengan menggunakan software Winlog Lite 2.06.74 secara jarak jauh.

4. Kinerja Alat :

- Otomatisasi pada setiap inkubator menggunakan pendeteksi suhu yaitu sensor suhu IC LM 35 yang difungsikan sebagai pendeteksi suhu $36^{\circ}\text{C} - 37^{\circ}\text{C}$ pada setiap inkubator.
- Sumber daya menggunakan tegangan 220 VAC dan 24 VDC

5. Dimensi alat sebagai simulasi sistem pengendalian suhu adalah :

- Inkubator yang terbuat dari *acrylic* dengan luas bangunnya :

Panjang : 70 cm

Lebar : 40 cm

Tinggi : 37.5 cm

BAB IV

PENGUJIAN SISTEM

4.1 Pendahuluan

Dalam bab ini membahas tentang pengujian dan pengukuran sistem dari peralatan yang dibuat. Secara umum pengujian ini bertujuan untuk mengetahui kinerja dari peralatan dan sistem yang telah yang telah direalisasikan dapat bekerja sesuai dengan perencanaan yang telah direncanakan.

4.2 Pengujian Sensor Suhu

4.2.1 Tujuan

Tujuan pengujian sensor suhu IC LM 35 adalah untuk mengetahui apakah rangkaian sensor yang digunakan sudah dapat bekerja dengan baik atau belum. Hal ini dapat diketahui dengan mengukur besarnya tegangan yang keluar saat sensor sedang membaca suhu. Sensor ini mempunyai prinsip kerja yaitu dengan merubah suhu kedalam tegangan, karena itu perlu dilakukan kalibrasi juga antara suhu dan tegangan yang keluar dari sensor tersebut. Sensor ini memberikan masukan ke *Smart relay* berupa input analog, berupa tegangan antara 0 – 10 V.

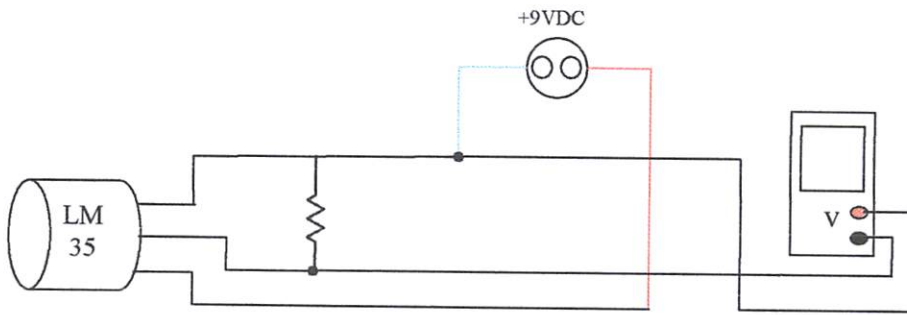
4.2.2 Peralatan yang Digunakan

Dalam pengujiannya sensor suhu IC LM 35 menggunakan peralatan sebagai berikut :

- Multimeter digital
- Rangkaian sensor suhu IC LM 35 yang terhubung dengan catu daya 14 VDC

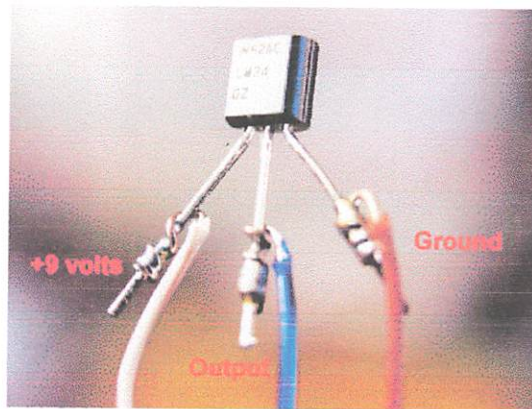
4.2.3 Prosedur pengujian

1. Membuat rangkaian sensor suhu IC LM 35 sesuai gambar 4.1 dibawah.
2. Memasang suplai tegangan 14 VDC pada rangkaian.
3. Mengukur suhu keluaran sensor IC LM 35 dengan menggunakan multimeter digital.

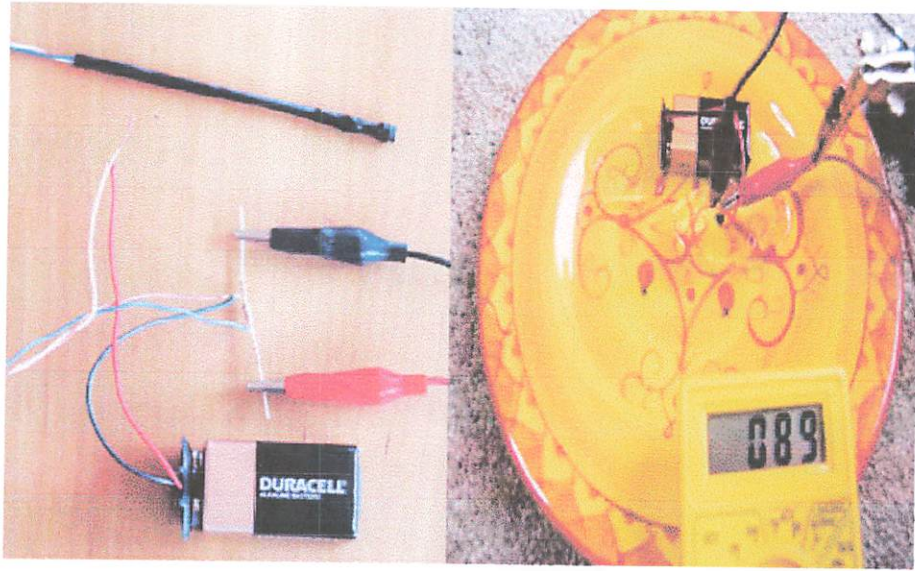


Gambar 4.1
Pengujian Rangkaian Sensor Suhu IC LM 35

4.2.4 Hasil Pengujian



Gambar 4.2
Rangkaian Pada IC LM35 ^[9]



Gambar 4.3
Hasil Pengujian Sensor IC LM 35 [9]

Dari gambar pengujian diatas dimana sensor suhu IC LM 35 diuji pada sebuah es dan diketahui bahwa sensor IC LM35 akan mengeluarkan suhu 8.9°C.

Berikut ini adalah tabel hasil pengujian rangkaian sensor suhu IC LM35 berdasarkan tegangan dan suhu.

Tabel 4.1 Hasil Pengujian Sensor Suhu IC LM35

Sensor Suhu IC LM35	Suhu °C	Tegangan mV	Logika	
			Hidup	Mati
IC LM35 Inkubator 1 sensor 1	± 28	280	1	0
	36	360	1	0
	37	370	0	1
IC LM35 Inkubator 1 sensor 2	± 28	280	1	0
	36	360	1	0
	37	370	0	1
IC LM35 Inkubator 2 sensor 1	± 28	280	1	0

	36	360	1	0
	37	370	0	1
IC LM35 Inkubator 2 sensor 2	± 28	280	1	0
	36	360	1	0
	37	370	0	1

Dari hasil pengujian rangkaian sensor suhu IC LM35 pada gambar dan juga tabel diatas, dapat diketahui bahwa sensor suhu IC LM35 akan bekerja atau mendeteksi suhu $\pm 28^{\circ}\text{C}$ atau suhu ruangan maka sensor akan berlogika "1" atau high yang akan mengirimkan signal ke *Smart Relay* sebagai inputan hingga sensor suhu IC LM35 mendeteksi suhu 37°C maka akan berlogika "0" atau low sebagai inputan ke *Smart Relay* yang telah diprogram.

4.3 Pengujian *Power Supply*

4.3.1 Tujuan

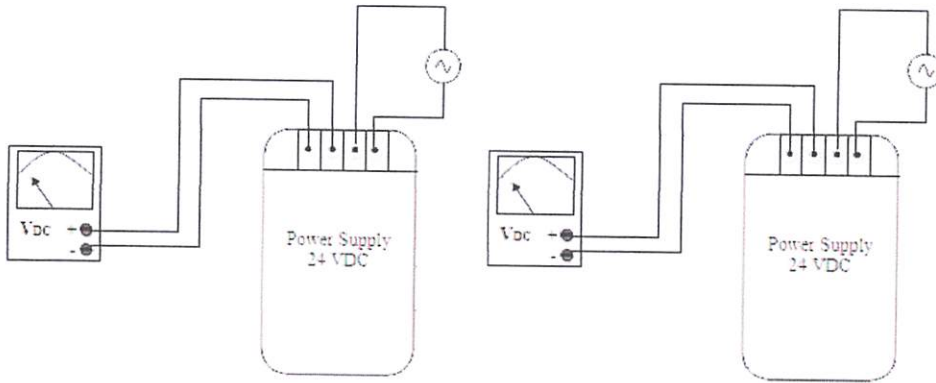
Untuk mengetahui apakah output dari rangkaian *power supply* sesuai dengan teggangan kerja yang dibutuhkan oleh *zelio smart relay*, *modbus* dan juga sensor suhu.

4.3.2 Peralatan yang Digunakan

- Power supply 24 VDC dan 14 VDC
- Multimeter Digital

4.3.3 Prosedur Pengujian

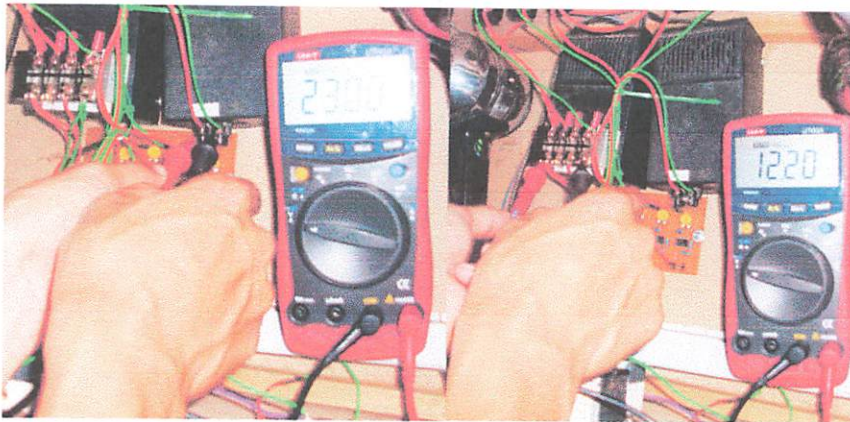
- Merangkai rangkaian pengujian *power supply* seperti gambar 4.4
- Melakukan pengukuran pada output *power supply*



Gambar 4.4
Rangkaian Pengujian *Power Supply* 24 VDC dan 14 VDC

4.3.4 Hasil Pengujian

Hasil pengukuran tegangan pada *power supply* ditunjukkan pada gambar dibawah :



Gambar 4.5
Hasil Pengujian Power Supply 24 VDC dan 14 VDC

4.4 Pengujian Daya Pada Sistem Kontrol

4.4.1 Tujuan

Tujuan pengujian daya pada sistem yaitu untuk mengetahui daya yang diperlukan oleh sistem secara keseluruhan.

4.4.2 Peralatan yang Digunakan

Dalam pengujian pada sistem menggunakan peralatan sebagai berikut :

- Multimeter digital.
- Tang *ampere*.

4.4.3 Prosedur Pengujian

1. Mengukur tegangan suplai PLN dengan menggunakan multimeter.
2. Mengukur arus yang mengalir pada sistem dengan menggunakan tang *ampere*.
3. Menghitung daya yang yang terpakai oleh sistem.

4.4.4 Hasil Pengujian

Hasil pengukuran arus dan tegangan pada sistem ditunjukkan pada gambar dibawah :



Gambar 4.6
Pengukuran Tegangan Suplai PLN



Gambar 4.7
Pengukuran Arus Yang Mengalir Pada Sistem

Dari gambar hasil pengukuran diatas, maka dapat diketahui daya yang diperlukan oleh sistem adalah :

$$\text{Rumus : } P = V \times I$$

Dimana : P = Daya yang diperlukan (Watt)

V = Tegangan (Volt)

I = Arus (Ampere)

$$\begin{aligned} P &= 218.6 \text{ Volt} \times 2.9 \text{ Ampere} \\ &= 633.94 \text{ Watt} \end{aligned}$$

4.5 Pengujian Keseluruhan Sistem

4.5.1 Tujuan

Tujuan pengujian keseluruhan sistem adalah untuk mengetahui seluruh sistem seperti komponen *Smart Relai Zelio Logic SR3*, Sensor Suhu, Software Winlog serta komponen yang lain berjalan dengan normal sesuai dengan perencanaan dan juga mengetahui apabila terjadi *error* pada sistem yang telah dibuat.

4.5.2 Peralatan yang Digunakan

Dalam pengujian sistem secara keseluruhan menggunakan peralatan sebagai berikut :

- Multimeter digital
- Rangkaian sensor suhu IC LM 35 yang terhubung dengan catu daya 9 VDC.
- Rangkaian keseluruhan sistem.
- Tang ampere

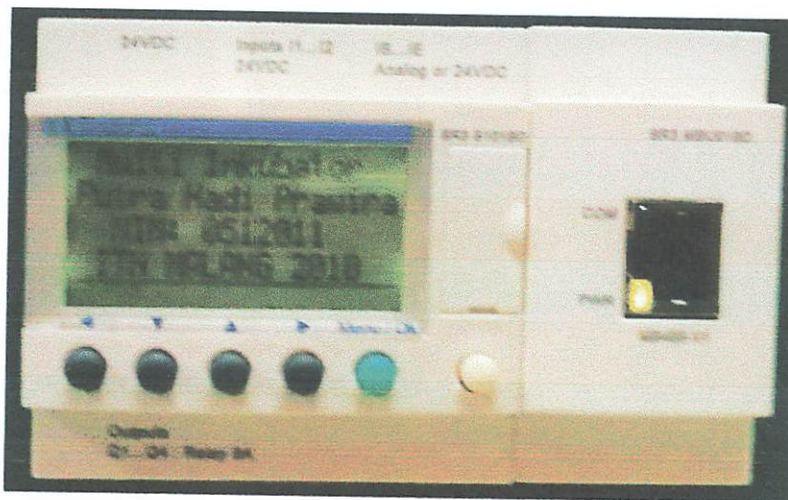
4.5.3 Prosedur Pengujian

1. Menghubungkan keseluruhan rangkaian sistem sesuai diagram blok.
2. Menghubungkan rangkaian kontrol ke sumber tegangan 1 fasa 220 Volt 50 Hz serta sumber 24 VDC.
3. Mentransfer program dari *Zelio Soft 2* kedalam *Smart Relay Zelio Logic SR3*.
4. Menghubungkan program Winlog dengan *Modbus Smart Relay Zelio Logic SR3*.
5. Run Program yang telah ditransfer kedalam *Smart Relay Zelio Logic SR3*.
6. Menekan saklar ON/OFF Smart Relay pada box panel untuk mengaktifkan inkubator 1 dan inkubator 2.
7. Menekan saklar 1 ON untuk mengaktifkan inkubator 1 serta sensor suhu primer dan back up dan *heater* (elemen panas).
8. Menekan saklar 2 ON untuk mengaktifkan inkubator 2 serta sensor suhu primer dan back up dan *heater* (elemen panas).

9. Menekan saklar 1 OFF untuk mematikan inkubator 1 serta sensor suhu primer dan back up dan *heater* (elemen panas).
10. Menekan saklar 2 OFF untuk mematikan inkubator 2 serta sensor suhu primer dan back up dan *heater* (elemen panas).
11. Membaca keadaan sensor suhu dan biarkan sistem berjalan sesuai program.
12. Membaca dan mengatur setingan inkubator melalui software winlog.

4.5.4 Hasil Pengujian

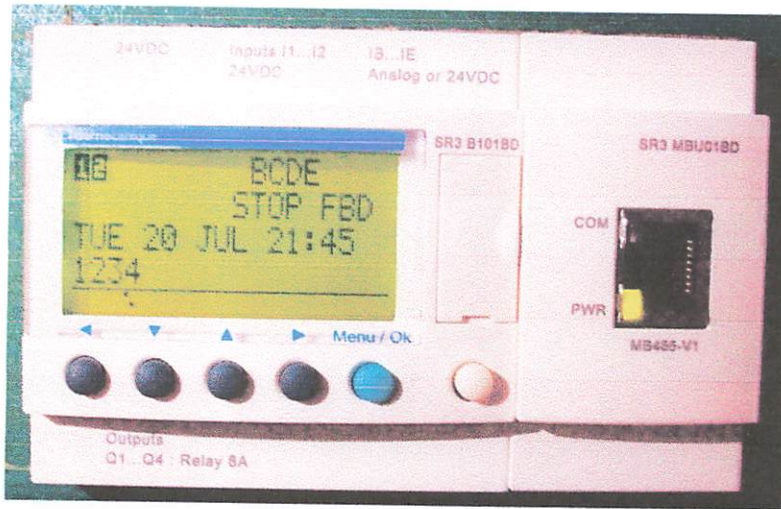
1. Untuk mengaktifkan sistem, dapat menekan tombol *push button On* atau lewat program *zelio soft 2* atau pun lewat modul *Smart Relay*. Jika pada *display Smart Relay* menampilkan nama maka sistem telah bekerja atau aktif tapi pada keadaan standby, seperti yang terdapat pada gambar dibawah ini :



Gambar 4.8
Gambar Proses *Smart Relay Zelio Logic SR3* Aktif/Standby.

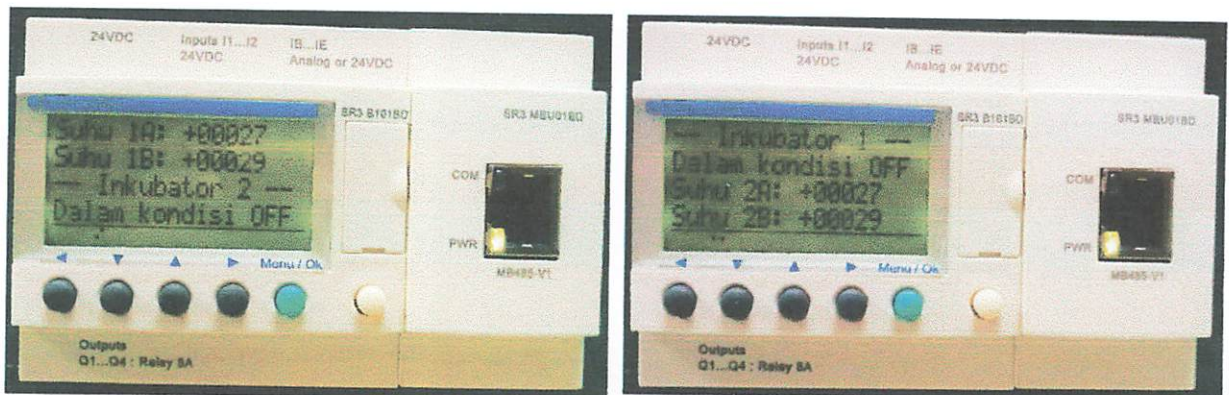
2. Untuk mematikan sistem, dapat menekan tombol *push button Off* atau menstop lewat program *zelio soft 2* atau pun lewat modul PLC. Jika pada display

Smart Relay menampilkan stop, maka sistem tidak bekerja seperti pada gambar dibawah berikut :



Gambar 4.9
Gambar Proses *Smart Relay Zelio Logic SR 3* Stop.

3. Untuk menjalankan sitem inkubator dengan cara menekan saklar On/Off pada panel box, seperti pada gambar :



Gambar 4.10
Gambar Proses *Smart Relay Zelio Logic SR3* Running/Bekerja.

Berikut ini adalah tabel hasil pengujian keseluruhan sistem dan sensor suhu (IC LM35) yang dipasang pada setiap inkubator dan juga pengendali jarak jauh (SCADA/software winlog):

Tabel 4.2 Hasil Pengujian Keseluruhan Sistem

No.	Kondisi Sistem	Respon Sistem
1	Sistem On	<ul style="list-style-type: none"> • <i>Smart Relay</i> pada panel menyala • Saklar sensor inkubator 1 dan inkubator 2 aktif • Sensor suhu IC LM35 dan <i>heater</i> tidak aktif • <i>Fan</i> (kipas) aktif • SCADA (software winlog) membaca kondisi inkubator 1 dan inkubator 2 On
2	Sistem Off	<ul style="list-style-type: none"> • <i>Smart Relay</i> pada panel mati • Saklar sensor inkubator 1 dan inkubator 2 aktif • Sensor suhu IC LM35 <i>heater</i> tidak aktif • <i>Fan</i> (kipas) tidak aktif • SCADA (software winlog) membaca kondisi inkubator 1 dan inkubator 2 Off
3	Saklar inkubator 1 On	<ul style="list-style-type: none"> • Sensor suhu IC LM35 primer dan back up inkubator 1 aktif • <i>Heater</i> (pemanas) aktif • <i>Fan</i> (Kipas) aktif • SCADA (software winlog) membaca kondisi inkubator 1
4	Saklar inkubator 2 On	<ul style="list-style-type: none"> • Sensor suhu IC LM35 primer dan back up inkubator 2 aktif • <i>Heater</i> (pemanas) aktif • <i>Fan</i> (Kipas) aktif • SCADA (software winlog) membaca kondisi inkubator 2
5	Saklar inkubator 1 Off	<ul style="list-style-type: none"> • Sensor suhu IC LM35 primer dan back up inkubator 1 tidak aktif • <i>Heater</i> (pemanas) tidak aktif • <i>Fan</i> (Kipas) aktif • SCADA (software winlog) tidak membaca kondisi inkubator 1
6	Saklar inkubator 2 Off	<ul style="list-style-type: none"> • Sensor suhu IC LM35 primer dan back up inkubator 2 tidak aktif • <i>Heater</i> (pemanas) tidak aktif • <i>Fan</i> (Kipas) aktif • SCADA (software winlog) tidak membaca kondisi inkubator 2
7	IC LM35 Primer	<ul style="list-style-type: none"> • Sensor akan membaca dari suhu ruangan $\pm 25^{\circ}\text{C}$ sampai 37°C, <i>heater</i> dan <i>fan</i> aktif

	Inkubator 1	<ul style="list-style-type: none"> • Sensor membaca suhu $>37^{\circ}\text{C}$, <i>heater</i> tidak aktif, <i>fan</i> aktif
8	IC LM35 Primer Inkubator 2	<ul style="list-style-type: none"> • Sensor akan membaca dari suhu ruangan $\pm 25^{\circ}\text{C}$ sampai 37°C, <i>heater</i> dan <i>fan</i> aktif • Sensor membaca suhu $>37^{\circ}\text{C}$, <i>heater</i> tidak aktif, <i>fan</i> aktif
9	IC LM35 Back Up Inkubator 1	<ul style="list-style-type: none"> • Sensor akan membaca dari suhu ruangan $\pm 25^{\circ}\text{C}$ sampai 37°C, <i>heater</i> dan <i>fan</i> aktif • Sensor membaca suhu $>37^{\circ}\text{C}$, <i>heater</i> tidak aktif, <i>fan</i> aktif
10	IC LM35 Back Up Inkubator 2	<ul style="list-style-type: none"> • Sensor akan membaca dari suhu ruangan $\pm 25^{\circ}\text{C}$ sampai 37°C, <i>heater</i> dan <i>fan</i> aktif • Sensor membaca suhu $>37^{\circ}\text{C}$, <i>heater</i> tidak aktif, <i>fan</i> aktif
11	Kondisi 00 Pada Smart Relay	<ul style="list-style-type: none"> • Pengontrolan inkubator 1 dan inkubator 2 dilakukan / diseting dari PC/Laptop melalui software winlog
12	Kondisi 01 Pada Smart Relay	<ul style="list-style-type: none"> • Pengontrolan inkubator 1 dan inkubator 2 dilakukan / diseting melalui <i>Smart Relay Zelio Logic</i>

Berdasarkan hasil pengujian keseluruhan sistem dan sensor suhu IC LM35 pada setiap inkubator dan juga pengendali jarak jauh (SCADA/software winlog) dapat diketahui :

1. Pada saat sistem on (aktif) atau dengan keadaan standby Smart Relay on dan sensor suhu IC LM35 dan heater off, yang dikarenakan inkubator 1 dan inkubator 2 terhubung oleh saklar on/off pada panel box yang berfungsi jika inkubator tersebut digunakan atau tidak.
2. Begitu pula sebaliknya pada saat sistem off (tidak aktif), maka respon sistem (Smart Relay) akan off.
3. Bila saklar on (aktif) maka inkubator 1 dan inkubator 2 On, sensor suhu IC LM35 baik yang primer maupun back up akan aktif dan sistem pun jalan.
4. Begitu pula sebaliknya jika saklar pada posisi off (tidak aktif) maka inkubator 1 dan inkubator 2 Off, sensor suhu IC LM35 baik primer maupun back up tidak aktif dan sistem pun tidak jalan.

5. Buzzer akan berfungsi apa bila sensor suhu kurang dari 18°C atau sensor suhu mengalami kerusakan atau sama dengan 0°C .

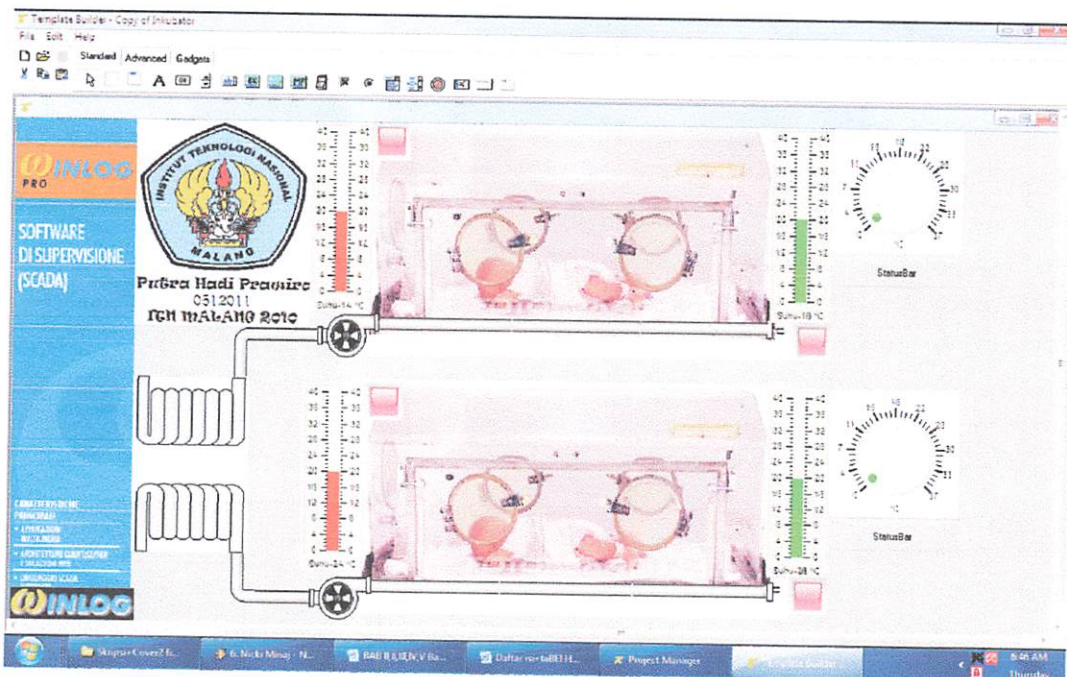


Gambar 4.11
Tampak samping Perancangan dan Pembuatan Sistem Pengendalian Multi Inkubator menggunakan PLC Smart Relay Zelio Logic Berbasis SCADA

6. Pada gambar diatas tampak sistem pengendali multi inkubator secara keseluruhan, pada setiap inkubator yang terdiri dari 2 sensor suhu IC LM35 yang diletakkan ditengah bagian belakang inkubator yang dimana kedua sensor tersebut berfungsi sama untuk mendeteksi suhu yang ada disekitar sensor atau didalam inkubator.
7. Pada saat pertama sistem on sensor suhu mendeteksi suhu sekitar $\pm 25^{\circ}\text{C}$ dan fan akan langsung bekerja, maka sistem akan merespon dengan menyalakan heater hingga suhu mencapai 37°C , setelah suhu tercapai 37°C maka heater akan mati dan suhu akan turun sampai setting yang ditentukan yaitu 36°C ,

kemudian heater akan bekerja lagi apabila sensor mendeteksi suhu $<36^{\circ}\text{C}$, begitu seterusnya sistem akan berjalan terus secara *continue*.

8. Apabila sistem gagal bekerja atau tidak berfungsi sebagai mana fungsinya masing-masing terutama sensor suhu maka buzzer akan berbunyi jika terjadi kegagalan sensor suhu tidak dapat mendeteksi suhu yang telah disetting atau sensor suhu rusak/tidak berfungsi.
9. Sistem pengendalian multi inkubator berbasis SCADA berfungsi sebagai monitoring dan juga mengontrol inkubator dari jarak jauh menggunakan kabel RJ45 yang dihubungkan dengan *modbus Smart Relay*, jika sistem dalam kondisi On maka SCADA (software winlog) akan membaca sensor suhu IC LM35 sedang mendeteksi suhu berapa pada inkubator. Pengaturan suhu yang dibutuhkan untuk suhu inkubator juga dapat dilakukan melalui SCADA (software winlog).



Gambar 4.12
Gambar kontrol SCADA (software winlog)

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan perencanaan, pembuatan, dan pengujian pada Sistem Pengendali Multi Inkubator Menggunakan PLC *Smart Relay Zelio Logic* Berbasis SCADA, dapat diambil beberapa kesimpulan sebagai berikut :

1. Pada Sistem Pengendali Multi Inkubator Menggunakan PLC *Smart Relay Zelio Logic* Berbasis SCADA ini, dibanding kan dengan inkubator yang ada dirumah sakit, dalam sistem kontrolnya terdapat perbedaan yaitu inkubator yang ada dirumah sakit dikontrol secara manual dan tidak dapat dimonitoring dan dikontrol melalui PC/Laptop.
2. keistimewaan sistem ini adalah sistem pengendalian multi inkubator ini disamping pengaturan suhu secara otomatis dan dilengkapi dengan program monitoring pada software *zelio soft versi 2* dan juga komunikasi SCADA menggunakan software winlog yang digunakan sebagai kontrol dan memonitor inkubator melalui Laptop/PC.

5.2 Saran

Pada alat hasil perancangan ini masih mempunyai kekurangan – kekurangan, untuk itu ada beberapa hal yang dapat dilakukan untuk pengembangan :

1. Agar sistem ini dapat bekerja dengan baik maka harus dilakukan pengecekan berkala pada sensor suhu IC LM35, karena sensor suhu IC

LM35 ini memiliki peranan yang paling penting dalam kinerja alat ini sebagai pendeteksi suhu didalam inkubator.

2. Untuk mengatasi apabila listrik PLN mati maka perlu digunakan catu daya lain sebagai back up power supply control sistem, baik untuk sensor suhu atau pun heater.
3. Dalam Pengembangan selanjutnya dapat menambahkan bukan hanya pengukuran suhu saja namun ditambahkan pula kelembaban.

DAFTAR PUSTAKA

- [1] Suhendar, Programmable Logic Controller, Graha Ilmu, 2005
- [2] Pelatihan SCADA Teknik Elektro ITN – Agustus 2009.pdf
- [3] Halim Kusumanegara, Tugas Akhir, Perancangan Dan Pembuatan Sistem Otomatisasi Kamar Hotel Berbasis Smart Relay Zelio Logic SR3, Institut Teknologi Nasional Malang, 2008
- [4] SR2 dan SR3 tech data (short).pdf
- [5] <http://digilib.its.ac.id/detil.php?id=4835&q=rancang%20bangun%20inkubator>
- [6] <http://agfi.staff.ugm.ac.id/blog/index.php/2009/01/apakah-scada-itu/>
- [7] <http://elektromedik.blogspot.com/2008/04/baby-incubator.html>
- [8] http://www.crompton-instruments.com/integra/us_int_serial_converter.pdf
- [9] <http://sulae.blogspot.com/2008/12/membuat-thermometer-digital-sederhana.html>
- [10] <http://digilib.petra.ac.id/jiunkpe/s1/elkt/2003/jiunkpe-ns-s1-2003-23496088-8811-termometer-chapter1.pdf>
- [11] http://4x4icon.com/offroad/headlight_warning/index.htm

Lampiran

1



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAM TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Putra Hadi Prawira
Nim : 05.12.011
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : **PERANCANGAN DAN PEMBUATAN SISTEM
PENGENDALI MULTI INKUBATOR
MENGUNAKAN PLC SMART RELAY ZELIO LOGIC
BERBASIS SCADA**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang program Starata Satu (S-1) pada :

Hari : Jumat
Tanggal : 20 Agustus 2010
Dengan nilai : 85,95 (A) *By*

PANITIA UJIAN SKRIPSI

Mengetahui,
Ketua Majelis Penguji


Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

ANGGOTA PENGUJI

Dosen Penguji I



Ir. Choirul Saleh, MT
NIP.P.1018800190

Dosen Penguji II



Bambang Prio Hartono, ST, MT
NIP.Y.1028400082



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAM TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

LEMBAR PERBAIKAN SKRIPSI

Nama Mahasiswa : Putra Hadi Prawira
NIM : 05.12.011
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Energi Listrik
Hari / Tanggal : Jumat / 20 Agustus 2010

No	Materi Perbaikan	Paraf
1.	- Kesimpulan Disempurnakan	

Telah Diperiksa / Disetujui:

Dosen Penguji II

Bambang Prio Hartono, ST, MT
NIP.Y.1028400082

Mengetahui

Dosen Pembimbing 1

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Dosen Pembimbing 2

Dr. Eng. Aryunto Soetedjo, ST, MT
NIP. Y. 1030800417

LEMBAR PERSETUJUAN

**PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI
MULTI INKUBATOR MENGGUNAKAN PLC SMART RELAY
ZELIO LOGIC BERBASIS SCADA**

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelara Sarjana Teknik Energi Listrik Strata Satu (S-1)*

Disusun Oleh :

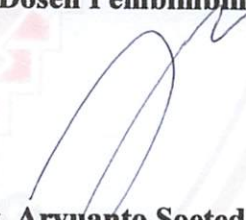
**PUTRA HADI PRAWIRA
NIM : 05.12.011**

Diperiksa dan Disetujui

Dosen Pembimbing 1

Dosen Pembimbing 2


Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800199


Dr. Eng. Aryunto Soetedjo, ST, MT
NIP. Y. 1030800417

**KOSENTRASI TEKNIK ENERGI LISTRIK
JURUSAN TEKNIK ELEKTRO S -1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010**



FORMULIR BIMBINGAN SKRIPSI

Nama : PUTRA HADI PRAWIRA
Nim : 0512011
Masa Bimbingan : 29 JANUARI – 29 JULI
Judul Skripsi : PERANCANGAN DAN PEMBUATAN SISTEM
PENGENDALI MULTI INKUBATOR MENGGUNAKAN PLC
SMART RELAY ZELIO LOGIC BERBASIS SCADA

No.	Tanggal	Uraian	Paraf Pembimbing
1.	03 – 05 – 2010	Konsultasi Software Winlog	
2.	10 – 05 – 2010	Konsultasi Software Zelio Soft 2	
3.	31 – 05 – 2010	Revisi Bab 1 dan Bab 2	
4.	14 – 06 – 2010	Revisi Bab 3	
5.	28 – 06 – 2010	Revisi Bab 4 dan Bab 5	
6.	12 – 07 – 2010	Revisi Tampilan Software Winlog	
7.	19 – 07 – 2010	Konsultasi Software Winlog Fix	
8.	26 – 07 – 2010	Revisi Penulisan Dan Gambar Makalah Seminar Hasil	
9.			
10.			

Malang, 04 Agustus 2010
Dosen Pembimbing,

Ir. Yusuf Ismail Nakhoda, MT

NIP.Y.1018800189



FORMULIR BIMBINGAN SKRIPSI

Nama : PUTRA HADI PRAWIRA
Nim : 0512011
Masa Bimbingan : 29 JANUARI – 29 JULI
Judul Skripsi : **PERANCANGAN DAN PEMBUATAN SISTEM
PENGENDALI MULTI INKUBATOR MENGGUNAKAN PLC
SMART RELAY ZELIO LOGIC BERBASIS SCADA**

No.	Tanggal	Uraian	Paraf Pembimbing
1.	03 – 05 – 2010	Konsultasi Software Winlog	
2.	10 – 05 – 2010	Konsultasi Software Zelio Soft 2	
3.	31 – 05 – 2010	Revisi Bab 1 dan Bab 2	
4.	14 – 06 – 2010	Revisi Bab 3	
5.	28 – 06 – 2010	Revisi Bab 4 dan Bab 5	
6.	12 – 07 – 2010	Revisi Tampilan Software Winlog	
7.	19 – 07 – 2010	Konsultasi Software Winlog Fix	
8.	26 – 07 – 2010	Revisi Penulisan Dan Gambar Makalah Seminar Hasil	
9.			
10.			


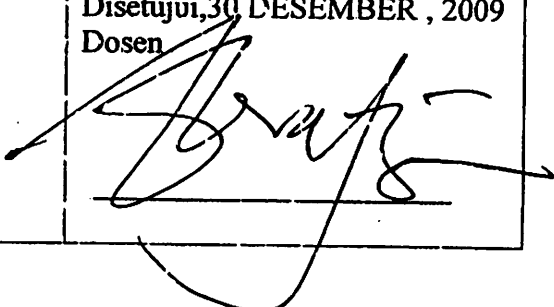
Malang, 04 Agustus 2010
Dosen Pembimbing,

Dr. Aryuanto Soetedjo, ST, MT

NIP. Y. 1030800417



FORMULIR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO/T. ENERGI LISTRIK S-1

1.	Nama Mahasiswa : PUTRA HADI PRAWIRA	Nim : 05.12.011		
2.	Waktu Pengajuan :	Tanggal	Bulan	Tahun
		30	DESEMBER	2009
	Spesifikasi Judul *)			
3.	a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri	e. Elektronika & Komponen f. Elektro Digital & Komputer g. Elektronika Komunikasi h. Lainnya		
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen *) :	Ketua Jurusan,  Ir. Yudi Limpraptono, MT NIP. Y. 1039500274		
5.	Judul yang diajukan Mahasiswa :	PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI MULTI INKUBATOR MENGGUNAKAN PLC SMART RELAY ZFLIO LOGIC BERBASIS SCADA		
6.	Perubahan Judul yang Disetujui Dosen materi bidang ilmu			
7.	Cacatan :			
8.	Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui, 30 DESEMBER, 2009 Dosen 		

Perhatian :

1. Formulir Pengajuan ini harap dikembalikan ke Jurusan paling lambat *satu minggu* setelah disetujui Kelompok Dosen Keahlian dengan dilampirkan Proposal Skripsi beserta persyaratan Skripsi sesuai **Form. S-1**.
2. *) dilingkari a, b, c,atau f, sesuai bidang Keahlian.
3. **) diisi oleh Jurusan.



Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak **Ir. Yusuf Ismail Nahkoda, MT**
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini :

Nama : Putra Hadi Prawira
Nim : 05.12.011
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Utama / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

**PERANCANGAN DAN PEMBUATAN SISTEM
PENGENDALI MULTI INKUBATOR
MENGUNAKAN PLC SMART RELAY ZELIO
LOGIC BERBASIS SCADA**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan Bapak kami ucapkan terima kasih.

Malang, 6 Januari 2010

Ketua
Jurusan Teknik Elektro S-1

Hormat kami,

Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274

Putra Hadi Prawira
Nim. 05.12.011

*) coret yang tidak perlu



Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak **Dr. Eng. Aryuanto Soetedjo, ST, MT**
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini :

Nama : Putra Hadi Prawira
Nim : 05.12.011
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing ~~Utama~~ / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

**PERANCANGAN DAN PEMBUATAN SISTEM
PENGENDALI MULTI INKUBATOR
MENGUNAKAN PLC SMART RELAY ZELIO
LOGIC BERBASIS SCADA**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan Bapak kami ucapkan terima kasih.

Malang, 6 Januari 2010

Ketua
Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274

Hormat kami,

Putra Hadi Prawira
Nim. 05.12.011

*) coret yang tidak perlu



PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : Putra Hadi Prawira
Nim : 05.12.011
Semester : 9 (sembilan)
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik

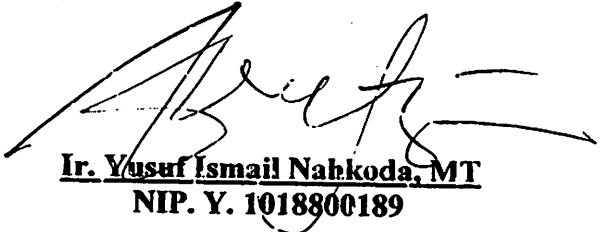
Dengan ini menyatakan bersedia / tidak bersedia *) membimbing Skripsi dari mahasiswa tersebut, dengan judul :

PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI MULTI INKUBATOR MENGGUNAKAN PLC SMART RELAY ZELIO LOGIC BERBASIS SCADA

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 7 Januari 2010

Kami Yang Membuat Pernyataan,



Ir. Yusuf Ismail Nahkoda, MT
NIP. Y. 1018800189

Catatan

Setelah disetujui agar formulir ini
Diserahkan mahasiswa yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut

*)coret yang tidak perlu

Form S-3b



INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus I : Jl. Bendungan Sigura-gura No.2
Kampus II: Jl. Raya Karanglo Km.2

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : Putra Hadi Prawira
Nim : 05.12.011
Semester : 9 (sembilan)
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik

Dengan ini menyatakan bersedia / tidak bersedia *) membimbing Skripsi dari mahasiswa tersebut, dengan judul :

PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI MULTI INKUBATOR MENGGUNAKAN PLC SMART RELAY ZELIO LOGIC BERBASIS SCADA

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 7 Januari 2010

Kami Yang Membuat Pernyataan,


Dr. Aryanto Soetedjo, ST, MT
NIP. Y. 1030800417

Catatan

Setelah disetujui agar formulir ini
Diserahkan mahasiswa yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut

*)coret yang tidak perlu

Form S-3b



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

PT BNI (PERSEF) MALANG
BANK NIAGA MALANG

Malang, 16 Juli 2010

Nomor : ITN- 774/ITA/2/10
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. DR. ENG. ARYUANTO S, ST, MT *)
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : PUTRA HADI P
Nim : 0512011
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

29 Januari 2010 s/d 29 Juli 2010

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1, apa bila lewat dari batas waktu tsb, maka skripsinya
digugurkan.

Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
Nip. Y. 1039500274

Tembusan Kepada Yth :

1. Mahasiswa Yang Berangkutan
2. Arsip

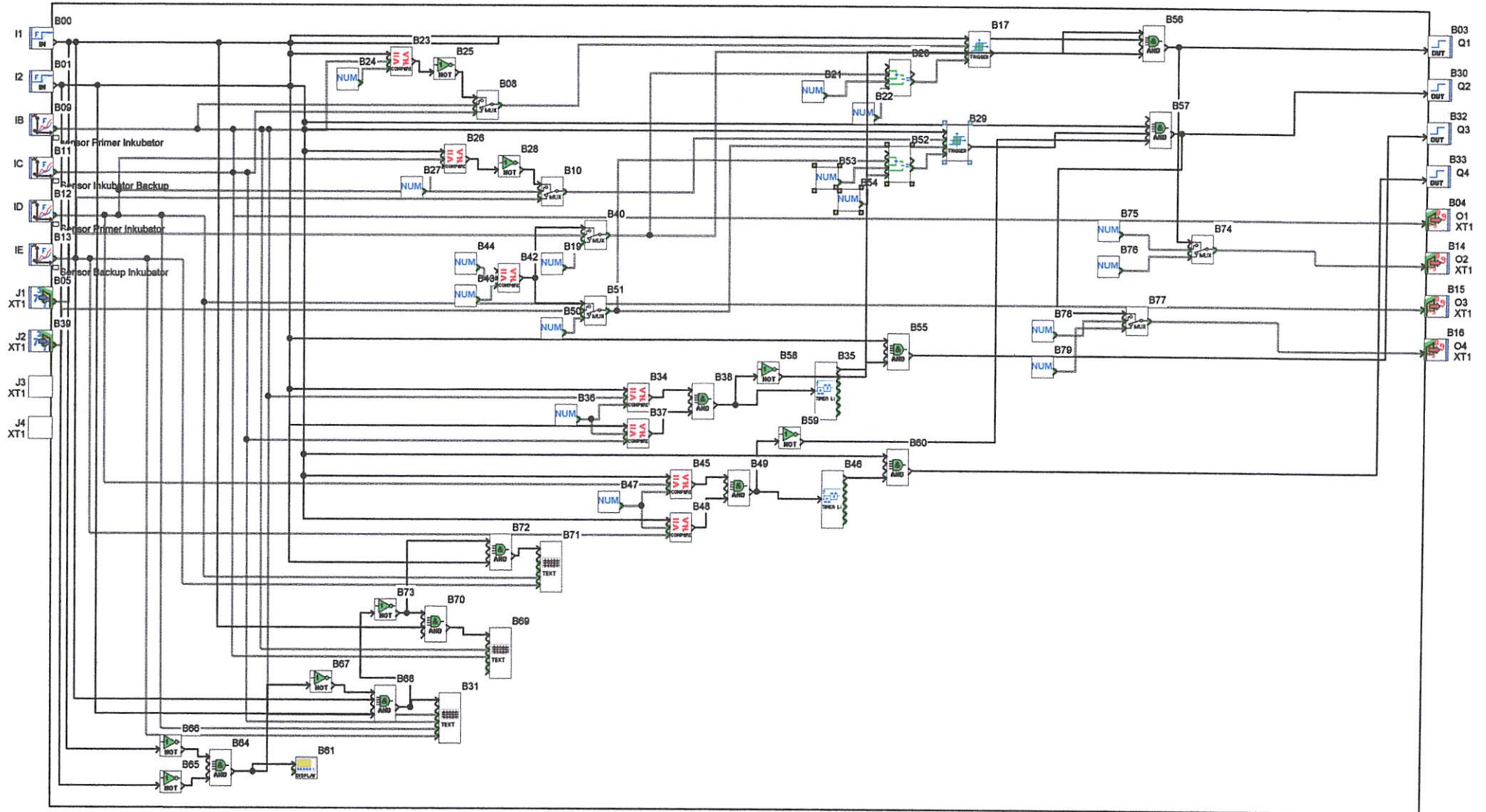
Form. S 4a

Lampiran

2



Program diagram





Program information

Author : Author
Project name : Title
Version : 0.0

<p>Module : SR3B101BD Sampling time in the module : 6 x 2 ms CHDOG action : Inactive Level of Hardware Input Filtering : Slow (3ms) Location of module front panel</p>	<p>XT1 : SR3MBU01 Number of wires and format : 2 wires, RTU Speed in bauds : 19200 Parity : None Modbus slave address : 1</p>
<p>Display format : dd/mm/yyyy Daylight Saving Time change activated : Europe Change to Daylight Saving Time : March, Last Sunday Change to winter time : October, Last Sunday</p>	



Physical inputs

Input	No	Symbol	Function	Lock	Parameters	Comment
	B00		Filtered discrete input	---	Duration of filtering : 1 x basic cycle (ms)	
	B01		Filtered discrete input	---	Duration of filtering : 1 x basic cycle (ms)	
	B09		Filtered analog input	---	Electrical connection at input : Potentiometer Cut-off frequency : 0.05 Hz	sensor Primer Inkubator 1
	B11		Filtered analog input	---	Electrical connection at input : Potentiometer Cut-off frequency : 0.05 Hz	Sensor Inkubator Backup 1
	B12		Filtered analog input	---	Electrical connection at input : Potentiometer Cut-off frequency : 0.05 Hz	Sensor Primer Inkubator 2
	B13		Filtered analog input	---	Electrical connection at input : Potentiometer Cut-off frequency : 0.05 Hz	Sensor Backup Inkubator 2

Physical outputs

Output	No	Symbol	Function	Comment
1	B03		Discrete output	
2	B30		Discrete output	
3	B32		Discrete output	
4	B33		Discrete output	

Configurable functions

Symbol	Function	Lock	Latching	Parameters	Comment
NUM	Numerical constant	No	---	Value of the constant : 37	
NUM	Numerical constant	Yes	---	Value of the constant : 0	
NUM	Numerical constant	Yes	---	Value of the constant : 2	
	Comparison of 2 values	---	---	VALEUR 1 < VALEUR 2	
NUM	Numerical constant	Yes	---	Value of the constant : 18	
	Comparison of 2 values	---	---	VALEUR 1 < VALEUR 2	
NUM	Numerical constant	Yes	---	Value of the constant : 18	
	Text	---	---	See details below	
	Comparison of 2 values	---	---	VALEUR 1 = VALEUR 2	



Symbol	Function	Lock	Latching	Parameters	Comment
TIMER Li	Cyclic timing	Yes	No	On time : 0H 0M 1S Off time : 0H 0M 1S Function Li - Continuous flashing	
NUM	Numerical constant	Yes	—	Value of the constant : 0	
COMPARE	Comparison of 2 values	—	—	VALEUR 1 = VALEUR 2	
COMPARE	Comparison of 2 values	—	—	VALEUR 1 = VALEUR 2	
NUM	Numerical constant	Yes	—	Value of the constant : 0	
NUM	Numerical constant	No	—	Value of the constant : 0	
COMPARE	Comparison of 2 values	—	—	VALEUR 1 = VALEUR 2	
TIMER Li	Cyclic timing	Yes	No	On time : 0H 0M 1S Off time : 0H 0M 1S Function Li - Continuous flashing	
NUM	Numerical constant	Yes	—	Value of the constant : 0	
COMPARE	Comparison of 2 values	—	—	VALEUR 1 = VALEUR 2	
NUM	Numerical constant	No	—	Value of the constant : 37	
NUM	Numerical constant	Yes	—	Value of the constant : 0	
NUM	Numerical constant	Yes	—	Value of the constant : 2	
DISPLAY	LCD display	—	—	See details below	
TEXT	Text	—	—	See details below	
TEXT	Text	—	—	See details below	
NUM	Numerical constant	Yes	—	Value of the constant : 0	
NUM	Numerical constant	Yes	—	Value of the constant : 1	
NUM	Numerical constant	Yes	—	Value of the constant : 0	
NUM	Numerical constant	Yes	—	Value of the constant : 1	

DISPLAY (LCD display)

DISPLAY	LCD display													
M	U	I	T	I	I	N	K	U	H	A	T	O	R	
t	r	a	.	H	a	d	i	P	r	a	w	i	r	a
T	N	M	.	A	5	1	2	0	1	1				
T	N	M	.	A	5	1	2	0	1	1				



FBD TEXT (Text)

TEXT											
h	u		1	A	.	*	*	#	4	*	*
h	u		1	R	.	*	*	#	5	*	*
h	u		2	A	.	*	*	#	6	*	*
h	u		2	R	.	*	*	#	7	*	*

ation not authorized for #4 B09 Filtered ana input
 ation not authorized for #5 B11 Filtered ana input
 ation not authorized for #6 B12 Filtered ana input
 ation not authorized for #7 B13 Filtered ana input

TEXT											
h	u		1	A	.	*	*	#	4	*	*
h	u		1	R	.	*	*	#	5	*	*

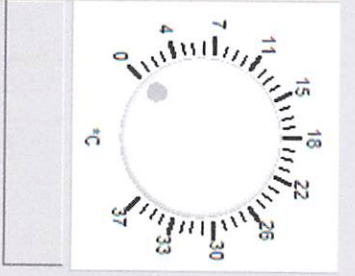
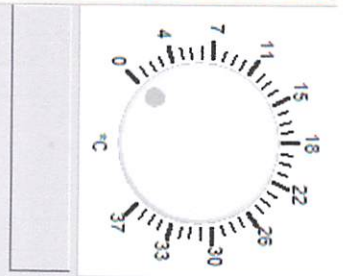
ation not authorized for #4 B09 Filtered ana input
 ation not authorized for #5 B11 Filtered ana input

TEXT											

ation not authorized for #6 B12 Filtered ana input
 ation not authorized for #7 B13 Filtered ana input



Putra Hadi Prasira
0512011
TGM MALANG 2010



LM35

Precision Centigrade Temperature Sensors

General Description

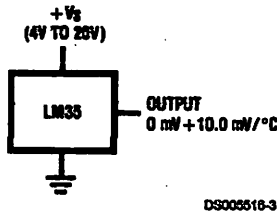
The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm 1/4^\circ\text{C}$ at room temperature and $\pm 3/4^\circ\text{C}$ over a full -55 to $+150^\circ\text{C}$ temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only $60\ \mu\text{A}$ from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55° to $+150^\circ\text{C}$ temperature range, while the LM35C is rated for a -40° to $+110^\circ\text{C}$ range (-10° with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

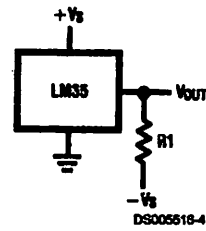
- Calibrated directly in ° Celsius (Centigrade)
- Linear $+ 10.0\ \text{mV}/^\circ\text{C}$ scale factor
- 0.5°C accuracy guaranteeable (at $+25^\circ\text{C}$)
- Rated for full -55° to $+150^\circ\text{C}$ range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than $60\ \mu\text{A}$ current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm 1/4^\circ\text{C}$ typical
- Low impedance output, $0.1\ \Omega$ for 1 mA load

Typical Applications



DS005516-3

FIGURE 1. Basic Centigrade Temperature Sensor
($+2^\circ\text{C}$ to $+150^\circ\text{C}$)



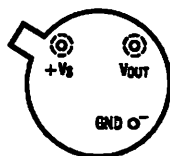
DS005516-4

Choose $R_1 = -V_S/50\ \mu\text{A}$
 $V_{\text{OUT}} = +1,500\ \text{mV}$ at $+150^\circ\text{C}$
 $= +250\ \text{mV}$ at $+25^\circ\text{C}$
 $= -550\ \text{mV}$ at -55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

Connection Diagrams

**TO-46
Metal Can Package***



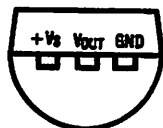
BOTTOM VIEW
DS006516-1

*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH

See NS Package Number H03H

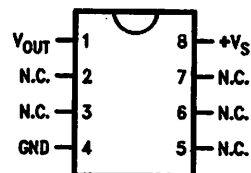
**TO-92
Plastic Package**



BOTTOM VIEW
DS006516-2

Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A

**SO-8
Small Outline Molded Package**

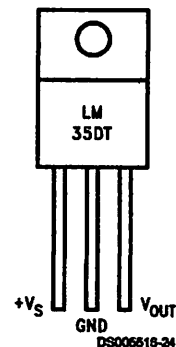


DS006516-21

N.C. = No Connection

Top View
Order Number LM35DM
See NS Package Number M08A

**TO-220
Plastic Package***



DS006516-24

*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	+35V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp.:	
TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-220 Package,	-65°C to +150°C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300°C

TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	-55°C to +150°C
LM35C, LM35CA	-40°C to +110°C
LM35D	0°C to +100°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	± 0.2	± 0.5		± 0.2	± 0.5		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	± 0.3			± 0.3		± 1.0	$^\circ\text{C}$
	$T_A = T_{MAX}$	± 0.4	± 1.0		± 0.4	± 1.0		$^\circ\text{C}$
	$T_A = T_{MIN}$	± 0.4	± 1.0		± 0.4		± 1.5	$^\circ\text{C}$
Linearity (Note 8)	$T_{MIN} \leq T_A \leq T_{MAX}$	± 0.18		± 0.35	± 0.15		± 0.3	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	$+10.0$	$+9.9,$ $+10.1$		$+10.0$		$+9.9,$ $+10.1$	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA	$T_A = +25^\circ\text{C}$ $T_{MIN} \leq T_A \leq T_{MAX}$	± 0.4 ± 0.5	± 1.0	± 3.0	± 0.4 ± 0.5	± 1.0	± 3.0	mV/mA mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$ $4\text{V} \leq V_S \leq 30\text{V}$	± 0.01 ± 0.02	± 0.05	± 0.1	± 0.01 ± 0.02	± 0.05	± 0.1	mV/V mV/V
Quiescent Current (Note 9)	$V_S = +5\text{V}, +25^\circ\text{C}$	56	67		56	67		μA
	$V_S = +5\text{V}$	105		131	91		114	μA
	$V_S = +30\text{V}, +25^\circ\text{C}$	56.2	68		56.2	68		μA
	$V_S = +30\text{V}$	105.5		133	91.5		116	μA
Change of Quiescent Current (Note 3)	$4\text{V} \leq V_S \leq 30\text{V}, +25^\circ\text{C}$	0.2	1.0		0.2	1.0		μA
	$4\text{V} \leq V_S \leq 30\text{V}$	0.5		2.0	0.5		2.0	μA
Temperature Coefficient of Quiescent Current		$+0.39$		$+0.5$	$+0.39$		$+0.5$	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature Rated Accuracy	In circuit of <i>Figure 1</i> , $I_L = 0$	$+1.5$		$+2.0$	$+1.5$		$+2.0$	$^\circ\text{C}$
Long Term Stability	$T_J = T_{MAX}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	$T_A = +25^\circ\text{C}$	± 0.4	± 1.0		± 0.4	± 1.0		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	± 0.5			± 0.5		± 1.5	$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$	± 0.8	± 1.5		± 0.8		± 1.5	$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$	± 0.8		± 1.5	± 0.8		± 2.0	$^\circ\text{C}$
Accuracy, LM35D (Note 7)	$T_A = +25^\circ\text{C}$				± 0.6	± 1.5		$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$				± 0.9		± 2.0	$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$				± 0.9		± 2.0	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.3		± 0.5	± 0.2		± 0.5	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+10.0	+9.8, +10.2		+10.0		+9.8, +10.2	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	± 0.4	± 2.0		± 0.4	± 2.0		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.5		± 5.0	± 0.5		± 5.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	± 0.01	± 0.1		± 0.01	± 0.1		mV/V
	$4\text{V} \leq V_S \leq 30\text{V}$	± 0.02		± 0.2	± 0.02		± 0.2	mV/V
Quiescent Current (Note 9)	$V_S = +5\text{V}, +25^\circ\text{C}$	56	80		56	80		μA
	$V_S = +5\text{V}$	105		158	91		138	μA
	$V_S = +30\text{V}, +25^\circ\text{C}$	56.2	82		56.2	82		μA
	$V_S = +30\text{V}$	105.5		161	91.5		141	μA
Change of Quiescent Current (Note 3)	$4\text{V} \leq V_S \leq 30\text{V}, +25^\circ\text{C}$	0.2	2.0		0.2	2.0		μA
	$4\text{V} \leq V_S \leq 30\text{V}$	0.5		3.0	0.5		3.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.7	+0.39		+0.7	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2.0	+1.5		+2.0	$^\circ\text{C}$
Long Term Stability	$T_J = T_{\text{MAX}}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

Note 1: Unless otherwise noted, these specifications apply: $-55^\circ\text{C} \leq T_J \leq +150^\circ\text{C}$ for the LM35 and LM35A; $-40^\circ\text{C} \leq T_J \leq +110^\circ\text{C}$ for the LM35C and LM35CA; and $0^\circ\text{C} \leq T_J \leq +100^\circ\text{C}$ for the LM35D. $V_S = +5\text{Vdc}$ and $I_{\text{LOAD}} = 50 \mu\text{A}$, in the circuit of Figure 2. These specifications also apply from $+2^\circ\text{C}$ to T_{MAX} in the circuit of Figure 1. Specifications in boldface apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is 400°C/W , junction to ambient, and 24°C/W junction to case. Thermal resistance of the TO-92 package is 180°C/W junction to ambient. Thermal resistance of the small outline molded package is 220°C/W junction to ambient. Thermal resistance of the TO-220 package is 90°C/W junction to ambient. For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in boldface apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and $10\text{mV}/^\circ\text{C}$ times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in $^\circ\text{C}$).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of Figure 1.

Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a $1.5 \text{ k}\Omega$ resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

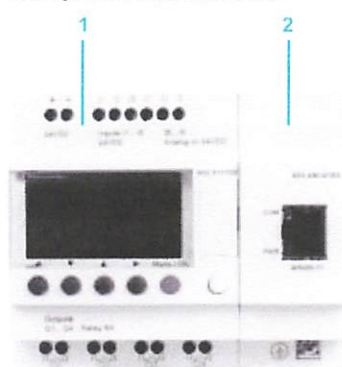
Zelio Logic smart relays

Modbus slave communication module

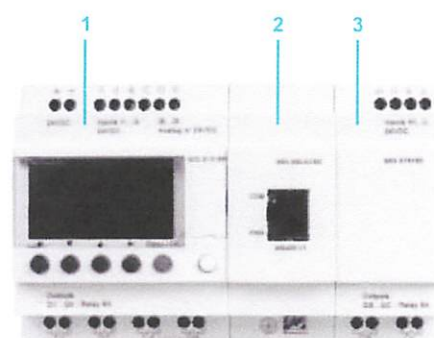


SR3 MBU01BD

Examples of combinations:



- 1 Modular smart relay (10 or 26 I/O)
- 2 Modbus slave communication module



- 1 Modular smart relay (10 or 26 I/O)
- 2 Modbus slave communication module
- 3 I/O extension module (6, 10 or 14 I/O)

Presentation

The Modbus protocol is a master/slave protocol.

Two exchange methods are possible:

■ Question/answer:

Questions from the master are addressed to a specific Modbus slave. The master waits for the answer to be returned by the slave polled.

■ Distribution:

The master distributes a message to all the slave stations on the bus. These stations execute the instruction without sending an answer.

Zelio Logic modular smart relays are connected to the Modbus network via the Modbus slave communication module. This module is a slave that is not electrically isolated.

The Modbus slave communication module must be connected to an SR3 B...BD modular smart relay, with a \approx 24 V supply.

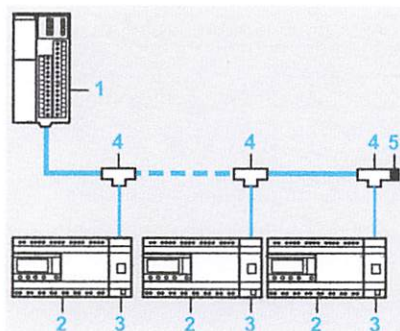
Configuration

The Modbus slave communication module can be configured:

- Independently, using the buttons on the smart relay (ladder language).
- On a PC, using "Zelio Soft" software (1).

When using a PC, programming can be carried out either in LADDER language, or in function block diagram language (FBD) (2).

Connection example



- 1 Modbus Master programmable controller (for example Twido).
- 2 Zelio Logic smart relay.
- 3 Modbus slave communication module.
- 4 T-junction.
- 5 Line terminator.

Description

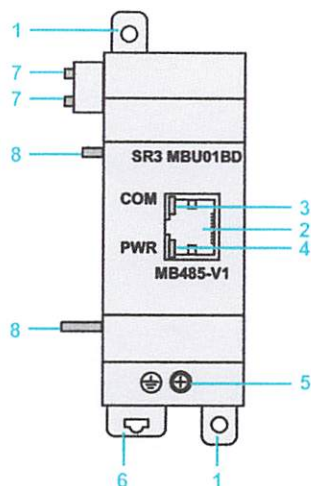
- The Modbus slave communication module is connected to a 2-wire or 4-wire Modbus network.
- The maximum length of the network is 1000 m (9600 bauds max., AWG 26).
- A maximum of 32 slaves may be connected to the Modbus network, or a maximum of 247 slaves with repeaters.
- Line end adaptors must be fitted to both ends of the line (10 nF/10 V, 120 Ω / 0.25 W in series).
- The line must be polarised (470 Ω / 0.25 W resistors) (3).
- The connection cable and its RJ45 male connectors must be shielded.
- The "COMMON" signal must be connected directly to the protective earth at one point on the bus.

(1) See page 14102/14.

(2) FBD: Function Block Diagram.

(3) The polarisation resistors must be managed by the master.

Description



Modbus slave communication module **SR3 MBU01BD** comprises:

- 1 Two retractable fixing lugs.
- 2 A Modbus network connection (RJ45 shielded female connector).
- 3 A communication LED (COM)
- 4 A Power ON LED (PWR)
- 5 A screw terminal block for the protective earth connection.
- 6 A spring clip for mounting on a 35 mm mounting rail.
- 7 Two locating pegs.
- 8 Two pins for clip-on fixing.

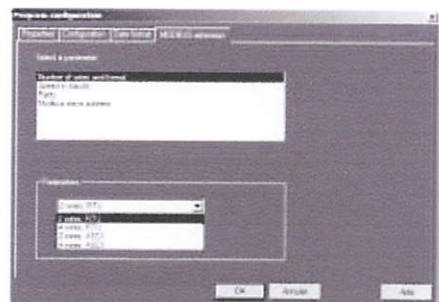
Environment characteristics

Product certifications		Pending: UL, CSA, GL, C-TICK
Conformity with the low voltage directive	Conforming to 73/23/EEC	EN 61131-2 (open equipment)
Conformity with the EMC directive	Conforming to 89/336/EEC	EN 61131-2 (Zone B) EN 61000-6-2, EN 61000-6-3 and EN 61000-6-4
Degree of protection	Conforming to IEC 60529	IP 20
Overvoltage category	Conforming to IEC 60664-1	3
Degree of pollution	Conforming to IEC/EN 61131-2	2
Ambient air temperature around the device	Operation	°C -20... +55 (+40 in enclosure), conforming to IEC 60068-2-1 and IEC 60068-2-2
	Storage	°C -40... +70
Maximum relative humidity		95 % without condensation or dripping water
Maximum operating altitude	Operation	m 2000
	Transport	m 3048
Mechanical resistance	Immunity to vibrations	Conforming to IEC 60068-2-6, test Fc
	Immunity to mechanical shock	Conforming to IEC 60068-2-27, test Ea
Resistance to electrostatic discharge	Immunity to electrostatic discharge	Conforming to IEC 61000-4-2, level 3
Resistance to HF interference (immunity)	Immunity to electromagnetic radiated fields	Conforming to IEC 61000-4-3, level 3
	Immunity to fast transients in bursts	Conforming to IEC 61000-4-4, level 3
	Immunity to shock waves	Conforming to IEC 61000-4-5
	Radio frequency in common mode	Conforming to IEC 61000-4-6, level 3
	Voltage dips and breaks (~)	Conforming to IEC 61000-4-11
	Immunity to damped oscillation waves	Conforming to IEC 61000-4-12
Conducted and radiated emissions	Conforming to EN 55022/11 (Group 1)	Class B

Zelio Logic smart relays

Modbus slave communication module

Parameter entry



Software workshop parameter entry window

Parameters can be entered either using Zelio Soft software, or directly using the buttons on the Zelio Logic smart relay.

When the "RUN" instruction is given, the Zelio Logic smart relay initialises the Modbus slave communication module in a configuration previously defined in the basic program.

The Modbus slave communication module has 4 parameters:

- number of UART wires and format of the frames on the Modbus network,
- transmission speed,
- parity,
- network address of the Modbus module.

The default parameter settings are as follows: 2-wire, RTU, 19 200 bauds, even parity, address n°1.

Parameter entry	Options
Number of wires	2 or 4
Frame format	RTU or ASCII
Transmission speed in bauds	1200, 2400, 4800, 9600, 19 200, 28 800, 38 400, 57 600
Parity	None, even, odd
Network address	1 to 247

Addressing of Modbus exchanges

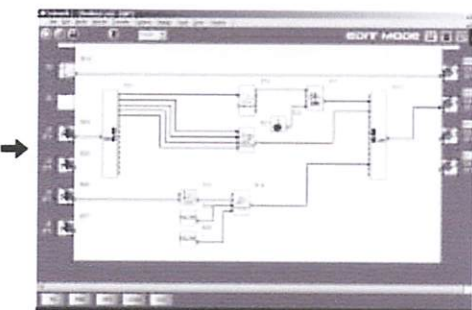
Function block diagram (FBD) programming (1)

In FBD mode, the 4 input data words (16 bits) (J1XT1 to J4XT1) and the 4 output data words (O1XT1 to O4XT1) can be accessed by the application. Dedicated function blocks make it possible to:

- break down a 'complete' type input (16 bits) into 16 separate 'bit' type outputs.
- Example: break down a Modbus type input (J1XT1 to J4XT1) and copy these status values to discrete outputs.
- make up a 'complete' type output (16 bits) from 16 separate 'bit' type inputs.
- Example: transfer the status value of the discrete inputs or the status of a function to a Modbus type output (O1XT1 to O4XT1).

Modbus exchanges	Code	Number of words
→	Read/Write 16, 06 or 03	4
⇨	Read 03	4
⌚	Read/Write 16, 06 or 03	4
⇨		
→		
Status ⇨	Read 03	1

(1) FBD: Function Block Diagram.



LADDER programming

In LADDER mode, the 4 data words (16 bits) to be exchanged cannot be accessed by the application. Transfers with the master are implicit and are effected in a way that is totally transparent.

Modbus exchanges	Code	Number of words
Image of smart relay I/O	Read 03	4
⌚	Read/Write 16, 06 or 03	4
⇨		
→		
Status ⇨	Read 03	1

Zelio Logic smart relays

Modbus slave communication module

References



SR3 MBU01BD

Modbus slave communication module

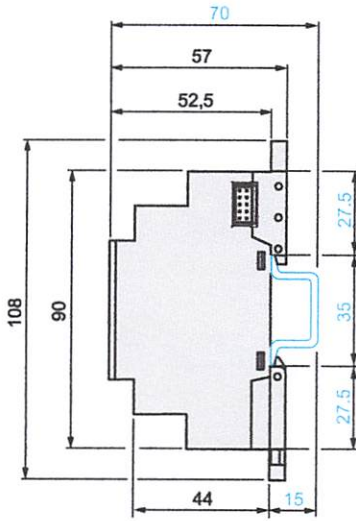
For use with	Reference	Weight kg
Modular smart relays SR3 BeeeBD	SR3 MBU01BD	0.300

Connection accessories

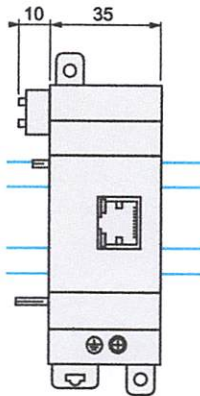
Description	Reference	Weight kg
T-junction complete with 0.3 m cable	VW3 A8 306TF03	-
T-junction complete with 1 m cable	VW3 A8 306TF10	-
T-junction without cable	170 XTS 04100	-

Dimensions and mounting

Side view



Rail mounting



Screw fixing

