

**APLIKASI SISTEM SCADA PADA PENGISIAN DAN  
PENUTUPAN KALENG SUSU SECARA OTOMATIS BERBASIS  
PLC SIEMENS S7-200 DAN S7-300**

**TUGAS AKHIR**

Diajukan untuk melengkapi salah satu syarat dalam menempuh gelar sarjana pada  
Jurusan Teknik Elektro Universitas Islam Indonesia

Yogyakarta



oleh :

Nama : Chairul Anwar

No. Mahasiswa : 02 524 050

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2007**

**LEMBAR PENGESAHAN PEMBIMBING**

**APLIKASI SISTEM SCADA PADA PENGISIAN DAN  
PENUTUPAN KALENG SUSU SECARA OTOMATIS BERBASIS  
PLC SIEMENS S7-200 DAN S7-300**



Pembimbing I

( Wahyudi Budi Pramono, ST )

Pembimbing II

( Yusuf Aziz Amrulloh, ST )

**LEMBAR PENGESAHAN PENGUJI**  
**APLIKASI SISTEM SCADA PADA PENGISIAN DAN**  
**PENUTUPAN KALENG SUSU SECARA OTOMATIS BERBASIS**  
**PLC SIEMENS S7-200 DAN S7-300**

**TUGAS AKHIR**

Disusun oleh :

Nama : Chairul Anwar

No. Mahasiswa : 02 524 050

**Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat**  
**Dalam Menempuh Gelar Sarjana Pada Jurusan Teknik Elektro**  
**Fakultas Teknologi Industri Universitas Islam Indonesia**

**Yogyakarta, Maret 2007**

Tim Penguji

Wahyudi Budi Pramono, ST.

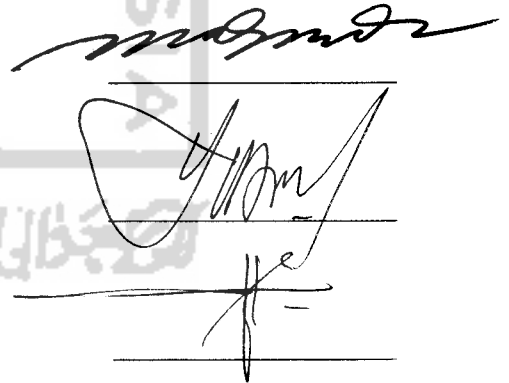
Ketua

Yusuf Aziz Amrulloh, ST.

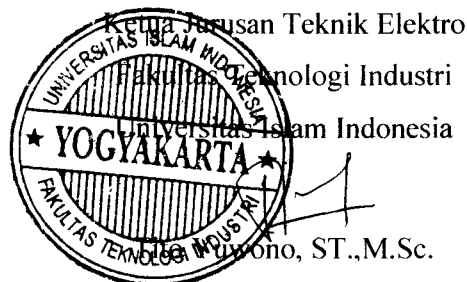
Anggota I

Medilla Kusriyanto, ST.

Anggota II



Mengetahui,



Ketua Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Universitas Islam Indonesia  
Wahyudi Budi Pramono, ST., M.Sc.



## Kalaman Persembahan

Kupersembahkan karya sederhana ini untuk:

Allah SWT.

Rosulullah Muhammad SAW.

Papa & Mama (Johan Safrie & Linawati) Tercinta atas do'a dan kasih sayangnya.

Kakak-kakakku Tercinta Mas Agung + Ayuk Yoyon dan Kak Belli + Ayuk Jujun.

Ponakanku yang Lucu Imam & Hani Smoga menjadi anak yang sholeh & sholehah

Special for My Luv ku Tersayang "Bhakti Dwi Harjanti" Smoga ALLAH SWT

Selalu senantiasa memberikan rahmat serta maghfiroh-Nya pada Kita.



## HALAMAN MOTTO

"Jadikanlah Sabar Dan Sholatmu Sebagai Penolongmu. Dan  
Sesungguhnya Yang Demikian Itu Sungguh Berat, Kecuali  
Bagi Orang-Orang Yang Khusyu"

( QS. Al Baqorah : 45)

"Barang Siapa Menempuh Jalan Untuk Mencari Ilmu, Maka  
Allah Memudahkan Jalan Baginya Menuju surga"

( HR. Muslim dan Abu Hurairah )

"Derajat Kemuliaan Seseorang Dapat Dilihat Dari Sejauh  
Mana Dirinya Punya Nilai Manfaat Bagi Orang Lain"

( HR. Bukhori )

"Pandanglah Kegagalan Sebagai Suatu Peluang Untuk  
Belajar, Sebagai Suatu Lompatan Kreatifitas, Sebagai

Suatu Kesempatan Untuk Menguji Gagasan Baru"

( Art Mortell )

## KATA PENGANTAR



*Assalamu 'alaikum Wr. Wb.*

Alhamdulillah, Segala puji hanya milik Allah SWT *rabb* semesta alam serta shalawat dan salam atas nabi Muhammad SAW. Atas rahmat dan taufik-Nya penulis dapat menyelesaikan tugas akhir dengan judul “**Aplikasi Sistem SCADA Pada Pengisian Dan Penutupan Kaleng Susu Secara Otomatis Berbasis PLC Siemens S7-200 Dan S7-300**” dapat diselesaikan dengan baik

Adapun maksud dan tujuan penyusunan tugas akhir ini adalah untuk melengkapi salah satu syarat dalam menempuh gelar sarjana pada Jurusan Teknik Elektro Universitas Islam Indonesia, Yogyakarta. Disamping itu untuk menambah pengetahuan terhadap ilmu yang telah dipelajari di bangku perkuliahan untuk dapat diterapkan di masyarakat.

Selama melakukan Tugas Akhir dan dalam penyusunan laporan ini, tidak lepas dari berbagai macam hambatan dan gangguan. Namun berkat motivasi, informasi dan konsultasi dari berbagai pihak, semua masalah dapat diatasi. Untuk itu penyusun menyampaikan rasa hormat sebagai ungkapan terima kasih kepada:

1. Bapak Fathul Wahid selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Tito Yuwono, ST, MSc selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Wahyudi Budi Pramono, ST. selaku dosen pembimbing I atas masukan dan nasehatnya.
4. Bapak Yusuf Aziz Amrulloh, ST. Selaku Sekretaris Jurusan dan dosen pembimbing II atas saran dan masukannya.
5. Bapak Ato Anshori Atas kesedian dan waktunya untuk membimbing penulis hingga mengerti.
6. Mas Iwan Rahman atas waktu, tempat, serta ilmu yang diberikan pada penulis. Semoga Allah SWT membalas semua amal sholeh yang telah dilakukan.
7. Mas Irwan atas waktu dan nasehat serta masukannya buat penulis.

8. Ibu Dwi Ana Ratna Wati, ST selaku Ka.Lab kendali dan Mikroprosesor atas kesediannya meminjamkan ruangan Lab serta fasilitasnya sebagai tempat penulis menyelesaikan tugas akhir ini.
9. Dosen dan karyawan Fakultas Teknologi Industri UII, Ka.Lab dan laboran jurusan Teknik Elektro yang telah banyak memberikan ilmu yang sangat bermanfaat bagi penulis.
10. Seluruh Staf Laboran Mas Heri, Mas Tri, Mas Agung yang telah amat sangat banyak membantu penulis.
11. Buat saudara-saudariku Alumni Takmir Masjid Masjid Ulil Alab UII atas semangat dan do'a nya.
12. Anak-anak kostku : Rezka, Mas Tung, Mas Asrofun, ST, Putra PH atas komputer dan printernya, Bambang, Yoga, Fahmie, serta Defri atas monitornya. Thanks banget buat kalian semua, semoga apa yang pernah kita lakukan bersama dapat menjadi kenangan yang terindah.
13. Seluruh mahasiswa jurusan Teknik Elektro UII dan U-UL CS yang telah memberi arti persahabatan selama ini. serta Ass. Lab Kendali dan Mikroprosesor.
14. Semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu hingga selesainya penyusunan laporan Tugas Akhir ini.

Penulis sangat menyadari, bahwa laporan Tugas Akhir ini masih jauh dari kesempurnaan. Baik dari segi isi, cara penyajian, serta teknik penulisan yang dipergunakan. Karenanya dengan segala kerendahan hati, penulis akan dengan senang hati untuk menerima dan mempertimbangkan segala bentuk saran dan kritik agar laporan ini dapat menjadi lebih baik, dan menuju kesempurnaan tentunya.

Besar harapan laporan ini dapat bermanfaat kepada penulis pada khususnya dan pembaca pada umumnya, amin.

*Wassalamu'alaikum Wr.Wb*

Jogjakarta, Maret 2007

Chairul Anwar

## ABSTRAK

Sistem otomatisasi di dunia industri sudah semakin berkembang dengan pesat. Hal ini dibuktikan dengan semakin banyaknya alat-alat produksi yang sudah tidak ditangani oleh tangan manusia, melainkan dikontrol dengan suatu sistem yang terprogram. Keseluruhan dari perancangan ini dikendalikan oleh PLC Siemens S7-200. Sedangkan untuk alamat sistem SCADA nya dipilih dari PLC Siemens S7-300 karena dapat digunakan oleh banyak *client* ataupun *user*. Adapun *interface* antara dua PLC yang digunakan yaitu sistem *profibus*. Pengisian dan penutupan kaleng susu secara otomatis ditunjukkan agar pengendalian dapat dibagi menjadi dua, yaitu pengendalian manual yang langsung dikendalikan dari tempat lokasi objek dan pengendalian SCADA yang dikendalikan dari operator serta dapat mengontrol dan memonitor seluruh pengendalian manual beserta objek yang dikendalikan. SCADA bekerja dengan memanfaatkan konektivitas dari modul ethernet yang terdapat pada PLC yang terhubung dalam suatu jaringan komputer, maka dari itu saat ini tidak menutup kemungkinan jika proses pengontrolan dan pengawasan dapat dilakukan lebih dari satu PC yang masih dalam satu jaringan. Fungsi *Controlling* pada pengisian dan penutupan kaleng susu ini adalah sebagai saklar, yaitu untuk menghidupkan dan mematikan objek. Sedangkan fungsi *monitoring* meliputi fungsi *display*, yaitu menampilkan jumlah kaleng susu tersebut. Dengan penelitian ini diharapkan sistem pengontrolan pengisian dan penutupan kaleng susu tidak harus di lokasi objek, melainkan dapat melalui komputer yang masih dalam satu jaringan bahkan bisa melalui internet. Dari pengujian yang dilakukan, sistem dapat berjalan sesuai dengan yang diinginkan yaitu sistem pengendalian melalui komputer yang masih dalam satu jaringan, sedangkan keseluruhan sistem dapat dikatakan berjalan dengan cukup baik

Kata kunci : PLC, Profibus, SCADA *software*.



## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN DOSEN PEMBIMBING.....	ii
LEMBAR PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERSEMBAHAN.....	iv
HALAMAN MOTTO.....	v
KATA PENGANTAR.....	vi
ABSTRAKSI.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	4
1.5 Metode dan Sistematika Penulisan.....	4
BAB II DASAR TEORI.....	7
2.1 PLC (Programmable Logic Controller).....	7
2.1.1 Pengenalan PLC.....	7
2.1.2 Flag.....	14
2.1.3 Perintah timer dan pencacahan.....	14
2.1.4 Cara Kerja PLC.....	16
2.2 PLC siemens S7-200.....	19
2.2.1 Central Processing Unit.....	19
2.2.2 Memori.....	19
2.2.3 Catu daya PLC.....	19

2.2.4 Modul masukan dan keluaran .....	20
2.2.5 Pemograman PLC .....	20
2.3 PLC siemens S7-300 .....	20
2.3.1 Rack .....	21
2.3.2 Power Supply .....	22
2.3.3 CPU .....	22
2.3.4 Konfigurasi Hardware .....	24
2.3.5 Software simatic step 7 .....	25
2.4 SCADA .....	29
2.5 Pemograman SCADA dengan WinCC .....	33
<b>BAB III PERANCANGAN SISTEM.....</b>	<b>35</b>
3.1 Perancangan Perangkat Keras .....	35
3.1.1 Motor Driver .....	37
3.1.2 Sensor .....	37
3.1.3 Sistem Pengisian .....	38
3.1.4 Sistem Pneumatik .....	38
3.1.5 Konfigurasi dua PLC .....	39
3.2 Profibus .....	40
3.3 Diagram Alir Sistem .....	41
3.3.1 Diagram Alir Sistem Pada Konveyor 1 .....	42
3.3.2 Diagram Alir Sistem Pada Konveyor 2 .....	43
3.4 Alamat PLC yang digunakan .....	44
3.4.1 Diagram ladder untuk start dan stop sistem .....	45
3.4.2 Diagram ladder pada konveyor 1.....	45
3.4.2.1 Diagram ladder pada pengisian kaleng .....	46
3.4.2.2 Diagram ladder untuk penempatan tutup kaleng .....	47
3.4.2.3 Diagram ladder penekan tutup kaleng .....	48
3.4.3 Diagram ladder pada konveyor 2 .....	48

3.5 Perancangan perangkat lunak .....	49
3.6 Jenis control yang digunakan .....	50
3.7 SUPERVISORY CONTROL AND DATA ACQUISITION ..	51
3.7.1 Prinsip kerja SCADA .....	51
3.7.2 Konektifitas <i>Link</i> SCADA .....	52
3.7.3 Pembuatan SCADA .....	53
3.7.3.1 Pembuatan project baru .....	53
3.7.3.2 Penambahan komunikasi driver .....	53
3.7.3.3 Pembuatan tag .....	57
3.7.3.4 Pembuatan tampilan proses .....	59
3.7.3.5 Pembuatan tombol .....	61
3.7.3.6 Mendefinisikan runtime .....	62
3.7.3.7 Waktu tunda yang digunakan .....	64
<b>BAB IV ANALISA DAN PEMBAHASAN .....</b>	<b>65</b>
4.1 Sistem komunikasi profibus .....	65
4.2 Standar komunikasi DP (Distributed Peripheral) .....	67
4.3 Indikator status EM 277 profibus-DP .....	68
4.4 Pengujian sistem pneumatik .....	69
4.5 Pengujian ketepatan posisi kaleng pada masing-masing sensor	70
4.6 Pengujian berat bubuk .....	71
4.7 Pemrograman SCADA .....	73
4.8 Analisa kerja sistem secara keseluruhan .....	76
<b>BAB V PENUTUP.....</b>	<b>78</b>
5.1 Kesimpulan.....	78
5.2 Saran.....	79

DAFTAR PUSTAKA

LAMPIRAN

## DAFTAR TABEL

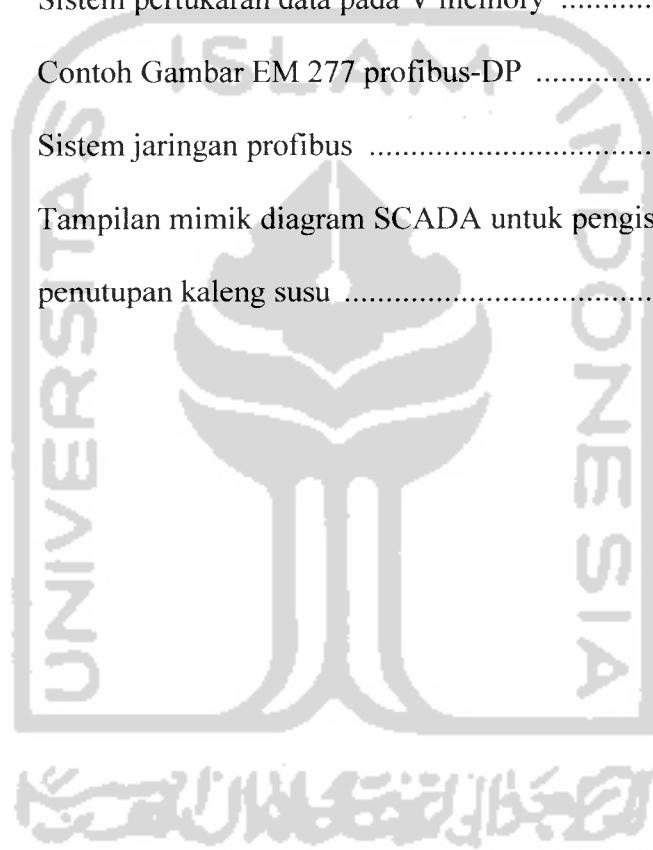
Tabel 2.1	Simbol Logika Kontak .....	13
Tabel 2.2	Urutan jumlah bit input dalam satu slot modul .....	23
Tabel 3.1	Alamat PLC Yang Digunakan .....	44
Tabel 4.1	Status LED Modul EM 277 Profibus-DP.....	68
Tabel 4.2	Pengujian Tekanan Sistem Pneumatik .....	69
Tabel 4.3	Pengujian Sensor Dengan Jarak Antara Setiap Kaleng Renggang (5 cm) .....	70
Tabel 4.4	Pengujian Sensor Dengan Jarak Antara Setiap Kaleng Rapat (2 cm) .....	71
Tabel 4.5	Pengujian Sensor Dengan Jarak Antara Setiap Kaleng Acak .....	71
Tabel 4.6	Pengujian Berat Bubuk .....	72

## DAFTAR GAMBAR

Gambar 2.1	Sebuah programable logic control .....	7
Gambar 2.2	Cara kerja PLC .....	17
Gambar 2.3	Posisi penempatan modul dalam sebuah rack .....	21
Gambar 2.4	New project pada simatic step 7 .....	25
Gambar 2.5	Insert simatic station 300 .....	25
Gambar 2.6	Tampilan simatic manager step 7 .....	27
Gambar 2.7	Konfigurasi hardware S7-300 .....	27
Gambar 2.8	Konfigurasi ethernet 1 .....	28
Gambar 2.9	Properties ethernet interface CP 343-1 lean .....	28
Gambar 3.1	Plant yang digunakan .....	35
Gambar 3.2	Blok diagram perancangan sistem .....	36
Gambar 3.3	Motor driver .....	37
Gambar 3.4	Sensor pendeteksi kaleng .....	37
Gambar 3.5	Ilustrasi sistem pneumatik .....	38
Gambar 3.6	Sistem pengiriman data pada V Memory .....	41
Gambar 3.7	Diagram alir sistem pada konveyor 1 .....	42
Gambar 3.8	Diagram alir sistem pada konveyor 2 .....	43
Gambar 3.9	Diagram ladder untuk start dan stop sistem .....	45
Gambar 3.10	Diagram ladder utama pada konveyor 1 .....	46

Gambar 3.11	Diagram ladder pada pengisian kaleng .....	47
Gambar 3.12	Diagram ladder untuk penempatan tutup kaleng ....	48
Gambar 3.13	Diagram ladder penekan tutup kaleng .....	48
Gambar 3.14	Diagram ladder pada konveyor .....	49
Gambar 3.15	Dua macam pengontrolan yang digunakan .....	50
Gambar 3.16	Penggunaan PLC bersama .....	51
Gambar 3.17	Pembuatan project baru .....	53
Gambar 3.18	Tampilan WinCC .....	54
Gambar 3.19	Penambahan komunikasi driver .....	54
Gambar 3.20	Pilihan penambahan komunikasi driver .....	55
Gambar 3.21	Penambahan komunikasi driver berdasarkan interface .....	55
Gambar 3.22	Connection properties .....	56
Gambar 3.23	Connection parameter – TCP/IP .....	56
Gambar 3.24	Hirarki tag management .....	57
Gambar 3.25	Penambahan tag group .....	57
Gambar 3.26	Tampilan pembuat tag .....	58
Gambar 3.27	Tampilan pembuatan alamat pada tag .....	58
Gambar 3.28	Tampilan pembuatan Tag telah selesai .....	59
Gambar 3.29	Pembuatan tampilan proses .....	60
Gambar 3.30	Push button NO .....	61
Gambar 3.31	Cara mengkonfigurasi tombol .....	62
Gambar 3.32	Tombol pada objek properties .....	62

Gambar 3.33	Definisi runtime pada <i>computer properties</i> di WinCC .....	63
Gambar 3.34	Tampilan runtime .....	63
Gambar 3.35	Penggunaan perputaran waktu .....	63
Gambar 4.1	V memory .....	66
Gambar 4.2	Sistem pertukaran data pada V memory .....	67
Gambar 4.3	Contoh Gambar EM 277 profibus-DP .....	68
Gambar 4.4	Sistem jaringan profibus .....	69
Gambar 4.5	Tampilan mimik diagram SCADA untuk pengisian dan penutupan kaleng susu .....	73



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sistem otomatisasi di dunia industri sudah semakin berkembang dengan pesat. Hal ini dibuktikan dengan semakin banyaknya alat-alat produksi yang sudah tidak ditangani oleh tangan manusia, melainkan dikontrol dengan suatu sistem yang terprogram. Salah satu sistem otomatisasi yang sering digunakan di dunia industri adalah menggunakan PLC (*Programmable Logic Control*).

Dunia pendidikan khususnya perguruan tinggi teknik sebagai salah satu pencetak tenaga ahli dalam industri dituntut untuk membekali mahasiswanya dengan keterampilan yang sesuai dengan bidangnya. Oleh karena itu diperlukan suatu sistem ataupun proses yang dapat memudahkan mahasiswanya untuk mempelajari hal yang berkaitan dengan bidang yang ditekuninya. Sebagai contoh pengoperasian PLC.

PLC merupakan salah satu kendali yang banyak digunakan di dunia industri. Berbagai mesin yang ada di industri banyak yang dikendalikan oleh PLC. Disamping mudah dalam perawatan dan pengolahan programnya, PLC juga sudah teruji keandalannya dibanding kendali yang lain.

Saat ini sangat dibutuhkan akses cepat untuk mendapatkan informasi-informasi yang akurat dalam menjalankan sebuah sistem kontrol tingkat tinggi. Hal inilah yang membuktikan betapa pentingnya sebuah teknologi yang disebut dengan *Human Machine Interface (HMI)*. *Human Machine Interface* yang berarti



mengontrol lajunya suatu proses dan tetap menjaga agar suatu sistem dapat bekerja dengan baik pada level maksimumnya. Yang mana teknologi ini amat sangat dibutuhkan dalam perkembangan industri. Suatu proses yang dijalankan dalam sebuah industri kini semakin berkembang dan semakin rumit, karena semakin tingginya kualitas produk yang akan dihasilkan, semakin banyak pula mesin-mesin yang digunakan, dan otomatis semakin banyak pula proses-proses yang dilakukan. Proses-proses tersebut bekerja dibawah pengawasan dari operator dan dituntut untuk mempunyai akses untuk mendapatkan informasi serta mengetahui kondisi sistem dari waktu ke waktu.

Dominasi yang kuat dari perusahaan-perusahaan raksasa yang ditunjang dengan teknologi tinggi membuat persaingan industri susu menjadi tidak seimbang, terlebih dikarenakan kemunculan industri-industri kecil (industri rumah tangga) tidak diiringi dengan teknologi yang memadai.

Banyaknya industri-industri kecil yang mengalami kegagalan dalam persaingan lebih disebabkan oleh animo masyarakat yang menganggap bahwa *kehigienisan* suatu produk ditunjang oleh tingkat kecanggihan proses produksi, dimana hal ini tidak dimiliki oleh sebagian besar industri-industri kecil yang ada.

Selain itu keterbatasan proses produksi yang disebabkan oleh masih banyaknya industri-industri kecil (industri rumah tangga) yang sebagian besar masih menggunakan peralatan manual guna menunjang proses produksinya.

Karena beberapa alasan di atas maka penulis berusaha untuk merancang pengisian dan penutupan kaleng susu menggunakan *SCADA* dengan memanfaatkan perangkat-perangkat yang ada di pasaran yang diharapkan dapat

berguna untuk menunjang pesatnya perkembangan otomatisasi industri belakangan ini.

Berdasarkan alasan-alasan diatas maka diangkat tema APLIKASI SISTEM SCADA PADA PENGISIAN DAN PENUTUPAN KALENG SUSU SECARA OTOMATIS BERBASIS PLC SIEMENS S7-200 DAN S7-300.

## 1.2 Rumusan Masalah

Dari latar belakang masalah itu, maka dalam pembuatan tugas akhir ini dapat dirumuskan sebagai berikut :

1. Bagaimana merancang sistem *SCADA* pada pengisian dan penutupan kaleng susu secara otomatis berbasis PLC siemen S7-200 dan S7-300 ?
2. Bagaimana cara merancang perangkat lunak sistem *SCADA* pada pengisian dan penutupan kaleng susu secara otomatis berbasis PLC siemens S7-200 dan S7-300 ?.

## 1.3 Tujuan Penelitian

Pembuatan tugas akhir yang berjudul “Aplikasi *SCADA* sistem pada pengisian dan penutupan kaleng susu secara otomatis berbasis PLC siemens S7-200 dan S7-300” mempunyai beberapa tujuan yaitu :

1. Untuk mengetahui aplikasi sistem *SCADA* pada pengisian dan penutupan kaleng susu.
2. Untuk merancang perangkat lunak sistem *SCADA* yang dapat digunakan sebagai monitor pengisian kaleng susu.

3. untuk mengetahui cara kerja sistem *SCADA* pada pengisian dan penutupan kaleng susu secara otomatis.

#### 1.4 Manfaat Penelitian

Dari pembuatan tugas akhir ini diharapkan alat ini dapat bermanfaat bagi masyarakat, negara dan ilmu pengetahuan dan teknologi. Adapun manfaat dari pembuatan tugas akhir ini adalah :

1. Bagi masyarakat.

Alat ini merupakan aplikasi nyata penerapan teknologi kendali digital kepada masyarakat untuk memperoleh keamanan serta kemudahan dalam penggunaannya.

2. Bagi negara.

Sebagai sumbangsih untuk mengurangi ketergantungan pada pihak asing atau luar negeri.

3. Bagi ilmu pengetahuan dan teknologi (IPTEK).

Agar terjadi perubahan pada kehidupan manusia menuju kearah yang lebih baik, terlebih perkembangan teknologi di dunia industri.

#### 1.5 Metode dan Sistematika Penulisan

Penulis melakukan beberapa tahap dalam pengerjaan termasuk pengumpulan data. Metode yang digunakan adalah sebagai berikut:

1. Metode Observasi

Melakukan pengamatan terhadap objek yang akan dibuat dengan melakukan percobaan-percobaan baik secara langsung maupun tak langsung.

## 2. Metode literature

Mencari dan mengumpulkan data-data tentang objek yang akan dibuat dari buku-buku ilmiah, majalah, laporan atau sumber lainnya.

## 3. Metode wawancara/interview

Setelah melalui metode dokumentasi dan observasi maka selanjutnya melakukan komunikasi dan tanya jawab dengan dosen pembimbing ataupun orang yang berpengalaman dibidangnya.

Untuk memudahkan sistematika penulisan, maka dibagi kedalam beberapa bab pembahasan dengan urutan sebagai berikut:

### **BAB I        PENDAHULUAN**

Bab ini akan menguraikan latar belakang dan alasan pemilihan judul, tujuan penulisan, pembatasan masalah, metodologi, dan sistem penulisan.

### **BAB II        TINJAUAN PUSTAKA**

Bab ini menguraikan teori tentang peralatan elektronik yang mendukung dalam perancangan sistem SCADA pada pengisian dan penutupan kaleng susu secara otomatis berbasis PLC siemens S7-200 dan S7-300.

**BAB III PERANCANGAN SISTEM**

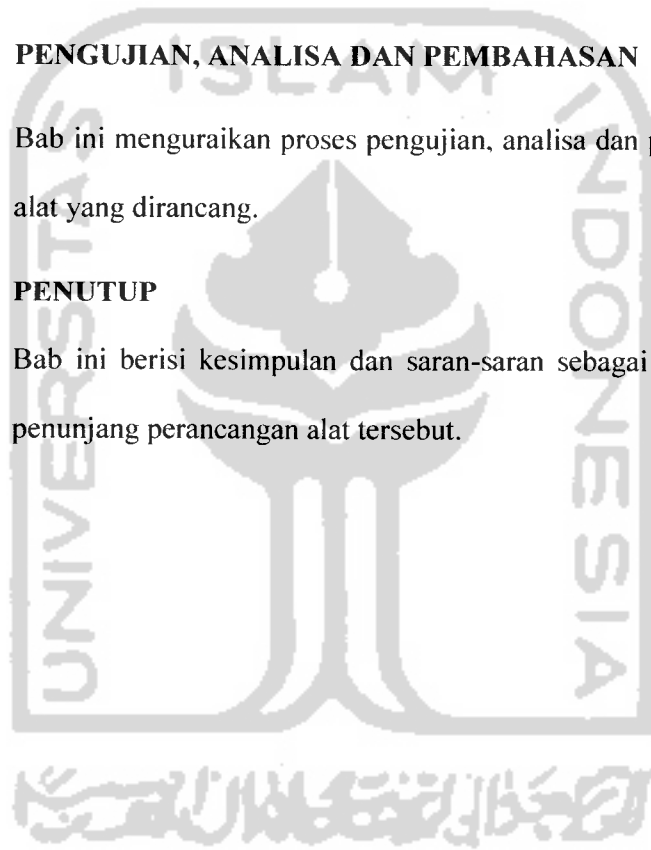
Bab ini berisi penjelasan tentang blok diagram rangkaian-rangkaian yang dipergunakan dalam perancangan sistem SCADA pada pengisian dan penutupan kaleng susu secara otomatis berbasis PLC siemens S7-200 dan S7-300 serta cara kerja dari alat tersebut.

**BAB IV PENGUJIAN, ANALISA DAN PEMBAHASAN**

Bab ini menguraikan proses pengujian, analisa dan pengamatan dari alat yang dirancang.

**BAB V PENUTUP**

Bab ini berisi kesimpulan dan saran-saran sebagai masukan untuk penunjang perancangan alat tersebut.



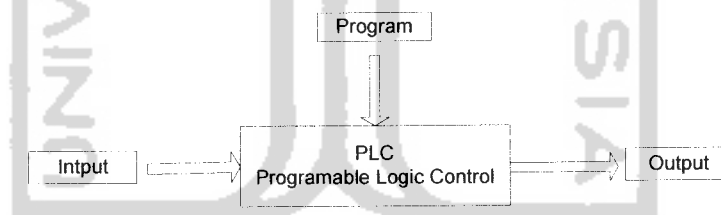
## BAB II

### DASAR TEORI

#### 2.1 PLC (Programmable Logic Control)

##### 2.1.1 Pengenalan PLC

*Programmable logic control* (PLC) merupakan suatu bentuk khusus pengontrol berbasis-mikroprosesor yang memanfaatkan memori yang dapat diprogram untuk menyimpan instruksi-instruksi dan untuk mengimplementasikan fungsi-fungsi semisal logika, *sequencing*, pewaktuan (*timing*), pencacahan (*counting*) dan aritmatika yang berguna untuk mengontrol mesin-mesin dan proses-proses yang dirancang untuk dioperasikan oleh para insinyur yang hanya memiliki sedikit pengetahuan mengenai komputer dan bahasanya.



Gambar 2.1 Sebuah programmable logic controller (PLC)

Pada awalnya PLC digunakan untuk menggantikan posisi relai karena disebabkan dengan berbagai permasalahannya, tetapi peningkatan lingkup fungsi didapatkan pada banyak aplikasi yang lebih kompleks. Pengontrolan yang dapat di program menawarkan beberapa keuntungan jika dibandingkan dengan jenis pengendali relai konvensional. Relai harus diberikan pengawatan kuat untuk melakukan fungsi khusus. Ketika sistem melakukan perubahan, pengawatan pun

melakukan fungsi khusus. Ketika sistem melakukan perubahan, pengawatan pun harus dirubah dan harus dimodifikasi, sehingga sangat memakan waktu. Belum lagi jika ada *error* atau ada kerusakan, maka untuk mencari lokasi kerusakan tersebut juga harus mencoba memeriksa satu-persatu pada kawat-kawat penghubung tersebut. Pengontrolan yang diprogram membatasi banyak pengawatan seperti pada rangkaian kontrol relai konvensional. Pengontrolan tersebut lebih mudah jika dibandingkan dengan pengontrolan yang didasarkan relai atau sejenisnya.

PLC terdiri atas tiga bagian yaitu :

1) CPU ( Central Processing Unit )

CPU adalah jantung dari sistem PLC. CPU didasarkan mikro-prosesor yang menggantikan relai pengendali, pencacah, timer, dan pembuat urutan *sequential*. Prinsip operasi CPU dapat dinyatakan sebagai berikut :

- a. CPU menerima data input dari berbagai sensor, mengeksekusi program, dan mengirimkan perintah ke bagian *output* yang tepat untuk mengendalikan piranti lainnya.
- b. Sumber daya arus searah diperlukan untuk menghasilkan tegangan level rendah yang digunakan prosesor dan modul I/O, daya ini dapat ditempatkan di CPU.
- c. Sebagian CPU mempunyai batere cadangan yang menjaga proses operasi tetap ada dalam penyimpanan, pada kejadian gagalnya daya yang ditempatkan.

Modul memori adalah bagian utama dari CPU. Memori berisi rencana pengendali atau unit penyimpanan *user program*. Informasi yang disimpan dalam memori berhubungan dengan cara data *input* dan *output* harus diproses. Secara *general* memori dari CPU ada beberapa tipe, yaitu:

a. RAM ( Random Acces Memory ).

Dirancang sehingga informasi dapat ditulis atau dibaca oleh memori. RAM menyediakan sarana yang bagus untuk menciptakan atau merubah program dengan mudah.

b. ROM ( Read Only Memory ).

Dirancang sedemikian rupa sehingga informasi yang disimpan di memori hanya dapat dibaca dan dibawah situasi yang biasa tidak dapat diganti.

c. EPROM ( Erasable Programable Read Only Memory ).

Yaitu ROM yang dapat diprogram dan setelah itu program tersebut secara permanen tersimpan didalamnya.

d. EEPROM ( Electrically Erasable Programable Read Only Memory ).

Merupakan memori yang menawarkan fleksibilitas pemrograman yang sama seperti RAM. Memori menyediakan penyimpanan program permanen, tetapi dapat diubah dengan menggunakan piranti program standar.

2) Bagian Input/Output (I/O)

Bagian I/O dari PLC terdiri dari modul *input* dan *output*. Sistem I/O membentuk *interface* dengan piranti lain yang dihubungkan pada



pengontrolan. Tujuan *interfacing* ini adalah untuk menyesuaikan kondisi berbagai sinyal yang diterima dari dan atau dikirimkan ke piranti eksternal. Piranti *input* seperti tombol tekan, sensor, dan lain-lain. Sedangkan piranti *output* seperti motor, starter motor, keran selenoid, lampu indikator, dan lain-lain.

Modul *interface input* menerima sinyal dari mesin atau piranti proses dan mengubahnya menjadi sinyal yang dapat digunakan oleh pengontrol. Sedangkan modul *interface output* mengubah pengontrol sinyal menjadi sinyal eksternal yang digunakan untuk mengendalikan mesin atau proses. Jenis modul *interface I/O* yang paling umum adalah jenis *discrete*, jenis *interface* ini menghubungkan piranti *input* dari sifat dasar *On/Off*. Modul *interface output analog* menerima data *digital* dari prosesor yang diubah pada tegangan yang berbanding lurus dengan arus untuk mengendalikan piranti medan *analog*. Data *digital* dilakukan pada rangkaian *converter* digital ke analog ( DAC ) untuk menghasilkan bentuk *analog* yang dibutuhkan.

Sinyal dihubungkan pada PLC melalui modul *input*. Modul *input* melakukan empat tugas pada sistem pengendali, yaitu :

- a. Merasakan sinyal yang diterima dari sensor pada *hardware*.
- b. Mengubah sinyal *input* menjadi *level* tegangan yang sesuai pada PLC tertentu.
- c. Mengisolasi PLC dari fluktuasi pada tegangan atau arus sinyal *output*.

- d. Memilih sinyal ke PLC yang menunjukkan sensor mana yang memulai sinyal.

Modul *output* memiliki fungsi yang sama dengan modul *input* kecuali dalam urutan pembalikan.

### 3) Piranti Pemrograman

Piranti pemrograman menyediakan sarana primer sehingga pemakai dapat berkomunikasi dengan rangkaian pengontrol yang dapat di program. Ini memungkinkan pemakai untuk memasuki, meng-edit, dan memonitor program yang terhubung ke unit prosesor dan mengijinkan akses ke memori pemakai.

Memprogram pengontrolan PLC perlu mempunyai pengetahuan bagaimana memorinya diorganisasikan. Secara umum dapat dibagi menjadi 3 bagian, yaitu : pengorganisasian memori berupa tabel data, program pemakai dan *memory house keeping*.

Tabel data menyimpan informasi yang diperlukan untuk melaksanakan program pemakai. Tabel data mencakup informasi misalnya status *input* dan *output*, nilai *timer* dan pencacah, penyimpanan data dan sebagainya. Isi tabel data dapat dibagi menjadi dua kategori, yaitu data status dan bilangan atau kode. Status adalah jenis *on/off* dari informasi yang disajikan dalam bentuk 1 atau 0, disimpan pada lokasi *bit* yang unik. Informasi bilangan atau kode diawali dengan kelompok *bit* yang disimpan pada lokasi *register* atau bahasa tertentu. Bilangan alamat memberikan perintah sesuai dengan *bit* status tertentu. *Bit* tersebut adalah




kemungkinan *ON* (logika 1) atau *OFF* (logika 0) yang menunjukkan apakah perintah *TRUE* atau *FALSE*.

Tabel data dapat dibagi menjadi tiga bagian sesuai dengan jenis informasi, yaitu tabel citra *input*, tabel citra *output*, dan *timer* serta penyimpanan cacah. Tabel citra *input* menyimpan status *input digital*, yang dihubungkan pada rangkaian *interface input*. Tabel citra *output* adalah susunan *bit* yang mengontrol status piranti *output digital* yang dihubungkan pada rangkaian *interface output*. Sedangkan tabel data yang terakhir menyimpan data *timer* dan hasil pencacahan. Program pemakai adalah tempat dimana diagram tangga logika yang diprogram, dimasukkan dan disimpan. Program pemakai akan menjelaskan memori total terbanyak dari sistem PLC tertentu. Program yang berisi logika yang diprogram pada format logika tangga. *Memori house keeping* digunakan untuk membawa fungsi-fungsi yang diperlukan untuk membuat prosesor beroperasi. Bagian ini tidak dapat diakses oleh pemakai.

Istilah bahasa pemrograman menunjuk pada metode pemakai yang mengkomunikasikan informasi pada PLC. Logika relai tangga adalah bahasa yang pertama kali dan paling populer yang dijumpai pada PLC. Logika relai tangga adalah bahasa pemrograman grafik yang dirancang untuk mewakili sedekat mungkin penampakan sistem relai yang diberi pengawatan. Logika tersebut menawarkan keuntungan besar untuk kontrol PLC. Tidak hanya *intuitif* yang dapat dipertanggung jawabkan, khususnya untuk teknisi dengan bekal pengalaman *relay*, tetapi juga efektif pada mode *on-line* ketika PLC benar-benar melakukan pengontrolan.

Bahasa pemrograman tangga pada dasarnya adalah suatu perangkat simbol dari perintah yang digunakan untuk menciptakan program pengontrol. Simbol perintah kontrol yang diinginkan pada memori PLC. Karena perintah penyetalan disusun dari simbol kontak-kontak, bahasa diagram tangga juga ditunjuk sebagai simbologi kontak. Perwakilan kontak dan kumparan adalah simbol pokok dari perangkat perintah diagram logika tangga. Tiga dari simbol yang digunakan untuk menterjemahkan logika kontak *relay* ke logika simbol kontak yaitu:

Tabel 2.1 Simbol logika kontak

Perintah	Ladder	STL
Examine On		A
Examine Off		AN
Output Energize		=

Selain itu terdapat juga logika simbol kontak yang menyatakan persyaratan logika digital **AND** dengan menghubungkan seri dua atau lebih simbol kontak diatas, atau logika digital **OR** dengan menghubungkan paralel dua atau lebih simbol-sombol kontak yang ada. Dan kedua logika *digital* ini dapat dipakai secara bersamaan seperti layaknya *wiring* kawat pada relai. Pada kondisi tertentu, *output* dibenarkan untuk berperan ganda menjadi *input* dan dapat dipakai berulang-ulang, yang mana nilainya akan terus sama sesuai dengan peran utamanya sebagai *output*. Akan tetapi *input* sendiri tidak bisa berperan ganda sebagai *output*, karna *input* itu sendiri adalah sebuah persyaratan tetapan logika yang akan menentukan nilai *output*.

### 2.1.1 Flag

*Flag* merupakan *memory address*, yaitu alamat I/O yang terdapat didalam memori PLC. Alamat dari *flag* ini biasanya menyatakan kondisi atau *input* dari *hardware* yang dikontrol, yang mana merupakan syarat input bagi satu atau beberapa kondisi atau *output* lainnya dalam sebuah rangkaian program. Dalam penulisan alamat *flag* (untuk Siemens PLC) harus diawali dengan lambang "M". *Flag* juga sama halnya dengan alamat I/O yang memiliki 8bit data, yaitu dari Mn.0, Mn.1, Mn.2, ....., Mn.7.

*Flag* hanya menyatakan kondisi (*digital*) dari *hardware* yang dikontrol, dapat juga berupa perintah atau *command* dari SCADA, tetapi tidak dapat berperan sebagai *output*. Jadi apabila alamat I/O pada PLC telah habis, maka untuk alamat-alamat *input* diatas dapat dituliskan di *memory address* sebagai *flag*.

### 2.1.2 Perintah timer dan pencacahan

*Timer* dan pencacah adalah perintah *output*. Pada *timer* terdapat tiga *bit* yang berperan dalam proses penundaan, diantaranya adalah:

1. *Bit Enable*. Adalah akan memiliki nilai 1 kapan pun perintah *timer* adalah *on*. Dan akan bernilai 0 apabila perintah *timer* adalah *off*.
2. *Bit Timer Timing*. Adalah berlogika 1 apabila tiap nilai yang diakumulasikan oleh *timer* selalu berubah, yang artinya *timer* sedang menetapkan waktu. Dan apabila nilai yang dialumulasikan tidak lagi berubah, maka *Bit Timer Timing* akan berlogika 0.

3. *Bit Done*. Berfungsi merubah logika *Bit Timer Timing* apabila nilai yang diakumulasikan telah mencapai nilai yang sudah ditetapkan sebelumnya.

Pada umumnya ada tiga timer, yaitu:

1. Timer Tunda On ( *Timer On Delay* )

*Timer on delay* atau sering disebut T-ON adalah timer yang paling umum digunakan. T-ON bekerja sedemikian rupa sehingga ketika anak tangga yang berisi *timer on*, maka *timer* periode *time out* dimulai. Pada akhir periode *time out*, *timer output* dibuat aktif. Jadi T-ON berfungsi untuk menunda keadaan *on* untuk waktu tertentu.

2. Timer Tunda Off ( *Timer Off Delay* )

*Timer Tunda Off* sering juga disebut dengan T-OFF. *Timer* ini bekerja dengan cara menyimpan nilai *output* 0 yang dihasilkan, maka akan terlihat pada jalur anak tangga seolah-olah tetap tersambung untuk beberapa saat. Jika kontinuitas logika hilang, maka *timer* akan memulai untuk mencacah interval waktu sampai dengan nilai waktu yang diakumulasikan sama dengan nilai yang telah ditetapkan dalam pemrograman. Dan apabila kondisi ini telah tercapai maka *output* akan di-*reset* kembali ke nilai 0.

3. *Retentive Timer On*

Berbeda dengan T-ON, RTO bekerja dengan menahan nilai yang diakumulasikan ketika anak tangga *timer FALSE*, dan akan melanjutkan penerapan waktu agar tetap *off* ketika pada saat anak

tangga *timer TRUE* lagi. Jadi pada dasarnya *timer* ini hanya bekerja pada satu kondisi saja, sedangkan pada satu kondisi lainnya nilainya tidak akan hilang melainkan tetap disimpan. *Timer* ini harus ditemani oleh perintah *reset* untuk meng-*set* nilai yang diakumulasikan oleh *timer* menjadi nol kembali.

Pencacah pada PLC mempunyai *format* pemrograman yang sama dengan *format timer*, tetapi ada perbedaan dalam cara kerjanya. *Timer* bekerja dengan menghitung waktu yang diberikan (di *set*) saat pemrograman dan setelah itu baru mengeluarkan *output*-nya. Sedangkan *counter* bekerja dengan mencacah jumlah *input* sampai dengan jumlah yang diberikan (di-*set*) saat pemrograman telah tercapai dan setelah itu barulah dihasilkan *output*. Jadi kesimpulannya *counter* adalah bergantung kepada *input*. Pencacahan PLC ada dua jenis, yaitu:

1. Perintah *Output* Pencacah Naik.

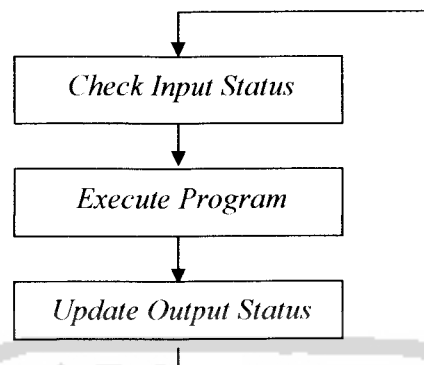
Perintah *output* pencacah naik akan menambahkan dengan satu setiap kejadian yang diitung terjadi.

2. Perintah *Output* Pencacah Turun.

Perintah *output* pencacah turun akan menghitung turun atau mengurangi dengan satu setiap kali kejadian yang dihitung terjadi.

### 2.1.3 Cara kerja PLC

Sebuah PLC bekerja dengan cara melakukan *scanning* terus-menerus secara kontinyu. *Scanning* dilakukan dalam tiga tahapan utama berikut:



Gambar 2.2 Cara kerja PLC

Tahap1 : Pertama kali PLC akan melakukan *scanning* terhadap seluruh *input* yang ada. PLC akan melihat setiap *input* untuk menentukan apakah dalam kondisi ON atau OFF. PLC merekam data ini dalam memorinya untuk digunakan pada tahap selanjutnya yaitu pada tahap eksekusi program.

Tahap2 : Selanjutnya PLC akan mengeksekusi program yang telah dimasukkan ke dalam PLC. Hasil program ini akan menentukan kondisi *output* pada saat *update status output*. Hasil eksekusi ini juga disimpan dalam memori untuk berbagai keperluan seperti *update output*, penghitungan, *timing*, dan lain sebagainya.

Tahap3 : Pada tahap terakhir, PLC akan meng-*update* status *output* berdasarkan hasil eksekusi program. Setelah proses *update* ini selesai, PLC akan kembali lagi ke Tahap1 dan mengulangi lagi ketiga tahap diatas terus-menerus.

Satu waktu *scan* didefinisikan sebagai waktu yang diperlukan untuk melakukan ketiga tahapan tersebut. Total waktu respon PLC merupakan hal yang harus



diperhatikan ketika akan menggunakan perangkat PLC. PLC memerlukan waktu untuk bereaksi terhadap suatu perubahan. Total waktu respon tersebut merupakan total waktu untuk merespon tiga hal, yaitu *input*, eksekusi, dan *output*.

PLC hanya dapat mengetahui *input on / off* pada saat melakukan *scanning input* (pada tahap *check input status*). Apabila suatu *input on* pada saat eksekusi program dan *off* sebelum *scanning* dimulai maka *input* tersebut tidak akan memberikan pengaruh terhadap PLC, karena PLC tidak sempat untuk melakukan *scan input* tersebut. Dengan demikian, untuk menjamin suatu input pasti akan terbaca oleh PLC, maka input tersebut harus *on* setidaknya selama satu selang waktu *checking input* ditambah satu waktu *scanning input*. Akan tetapi tidak ada jaminan suatu input akan berada pada kondisi ON dalam selang waktu yang sedemikian lama. Oleh karena itu ada dua cara untuk mengatasi masalah ini sehingga input dapat dibaca oleh PLC, yaitu :

1. *Pulse Stretch Function*

Fungsi ini akan memperpanjang sinyal input sehingga PLC dapat melihat *input* tersebut pada *scanning* berikutnya.

2. *Interrupt Function*

Fungsi ini akan melakukan *interrupt scanning* yang sedang dilakukan untuk melakukan instruksi khusus yang telah di program. Begitu input PLC ON, PLC akan segera mengeksekusi fungsi khusus ini. Setelah dieksekusi, proses *scanning* dilanjutkan kembali mulai titik dimana *interrupt* tadi dilakukan.

## 2.2. PLC siemen S7-200

### 2.2.1 *Central Processing Unit (CPU)*

*CPU* merupakan otak dari PLC yang berfungsi untuk menangani komunikasi dengan piranti eksternal, interkoneksi antar bagian-bagian internal PLC, eksekusi program, manajemen memori, dan memberikan sinyal ke keluaran.

### 2.2.2 Memori

Memori merupakan daerah CPU yang digunakan untuk tempat penyimpanan data pada PLC.

Kapasitas memori pada PLC sangat bervariasi tergantung model dari PLC tersebut. Pada Siemens S7-200, PLC type ini memiliki kapasitas memori 8192 bytes untuk menyimpan program, 5120 byte untuk memori data. Kapasitas memori ini tergantung penggunaannya dan seberapa jauh dalam mengoptimalkan ruang memori PLC, yang berarti berapa banyak penggunaan lokasi yang diperlukan program kontrol untuk mengendalikan *plant* tertentu.

### 2.2.3 Catu daya PLC

Catu daya PLC dapat diketahui dari jenis konfigurasi catu daya yang tertulis pada PLC. Jenis catu daya PLC ditulis dengan bentuk: jenis catu daya PLC/jenis catu daya masukan/jenis catu daya keluaran. Pada penelitian ini menggunakan PLC Siemens S7-200 CPU 224 yang mempunyai konfigurasi catu daya AC/DC/Relay, artinya PLC Siemens S7-200 CPU 224 menggunakan catu

daya AC, catu daya masukan DC dan catu daya keluaran AC atau DC serta PLC Siemens S7-300 CPU 313C-2DP.

#### **2.2.4 Modul masukan dan keluaran**

Modul masukan dan keluaran merupakan peralatan atau perangkat elektronik yang berfungsi sebagai perantara atau penghubung (*interface*) antara CPU dengan peralatan dari luar.

Unit masukan merupakan bagian yang diperlukan agar PLC dapat berhubungan dengan bagian kontrol proses. Unit masukan menerima sinyal dari kabel yang dihubungkan dengan peralatan masukan seperti sensor, saklar atau transduser. Modul keluaran merupakan peralatan-peralatan yang digunakan untuk mengeluarkan data-data yang telah diproses oleh CPU ke alamat keluaran yang ditentukan pengguna. Modul keluaran menyediakan tegangan keluaran untuk aktuator atau indikator alat.

#### **2.2.5 Pemrograman PLC**

Secara umum sistem pemrograman PLC dapat dilakukan dengan pembuatan diagram *ladder*. PLC Siemens S7-200 memiliki perangkat lunak khusus untuk memprogramnya, yakni STEP 7 –Micro/WIN.

### **2.3. PLC siemens S7-300**

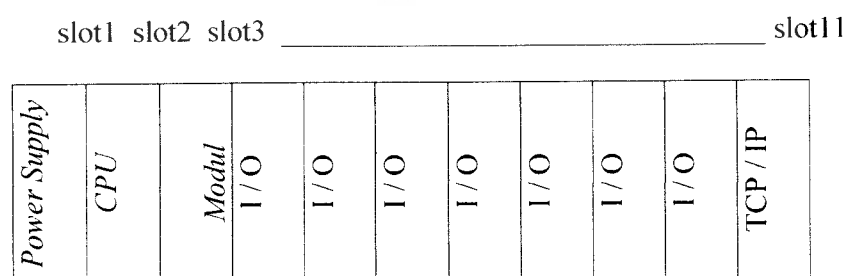
Pada prinsipnya PLC Siemen S7-200 memiliki persamaan dengan PLC Siemen S7-300, hanya saja pada penggunaan clientnya lebih sedikit pada S7-200

yaitu maksimal 2 client. Sedangkan pada PLC Siemen S7-300 dapat digunakan lebih dari 2 client.

### 2.3.1 Rack

*Rack* digunakan untuk menampung dan menghubungkan semua modul yang digunakan. Sebuah PLC, atau yang biasa disebut dengan *station* ini bisa saja berdiri atas beberapa *rack*. Rata-rata sebuah *rack* mampu menampung sebelas modul (slot), namun jika modul yang digunakan lebih dari itu, maka diperlukannya *rack* tambahan yang disebut juga dengan *expansion rack*, sedangkan *rack* utama tadi dinamakan dengan *central rack*. Perbedaan utama antara *expansion rack* dengan *central rack* adalah CPU, pada *central rack* terdapat sebuah CPU yang akan mengatur seluruh modul juga termasuk modul-modul yang berada di *expansion rack* dan menjalankan program, sedangkan di *expansion rack* tidak ditemukan lagi adanya CPU.

Dari ke sebelas modul yang terdapat didalam sebuah *rack*, haruslah sesuai penyusunannya, ini penting sebagai konfigurasi *hardware* pada saat pemrograman pada *software* SIMANTIC STEP7. berikut adalah pembacaan susunan modul yang dibenarkan dalam sebuah *rack* oleh SIMANTIC STEP7:



Gambar 2.3 Posisi penempatan modul dalam sebuah rack

### 2.3.2 Power supply

*Power Supply* digunakan untuk memberikan tegangan bagi CPU dan modul-modul I/O. pemakaian *power supply* harus memperhatikan CPU yang digunakan, karena ada beberapa CPU yang memiliki batas arus yang digunakan. *Power Supply* mengubah masukan tegangan bolak – balik menjadi tegangan searah yang akan di distribusikan ke seluruh modul – modul yang ada.

### 2.3.3 CPU

CPU merupakan pusat pemrosesan data dan mengatur seluruh operasi PLC mulai dari lalu lintas data antara CPU dan memori, I/O sampai menjalankan program yang ada didalam memori :

#### 1. Memori

Memori adalah rencana pengendali atau informasi yang disimpan pada pengontrol. Pada PLC Siemens S7-300 ini memorinya berupa *memory card*, yang bentuknya mirip seperti MMC, SDC, atau memori-memori sejenis. Pada *memory card* inilah semua data proses yang telah di *download* disimpan. selain bentuknya yang kecil dan tipis, tipe memori seperti ini mempunyai keunggulan, yaitu data yang tersimpan pada memori akan tetap utuh tersimpan apabila *power supply* terputus walau dalam waktu yang lama sekalipun, dan CPU akan kembali bekerja seperti sedia kala saat *power supply* terhubung kembali. *Memory card* yang digunakan berkapasitas 2MB, dan seperti layaknya memori yang bersifat external, kapasitas memori ini dapat di *up-grade* menjadi lebih besar lagi sesuai dengan kebutuhan pemrograman. Dan

sebagai catatan, PLC Siemens S7-300 ini tidak dapat bekerja apabila *memory card* dicabut.

## 2. I/O

Bagian I/O dari PLC adalah bagian yang terdiri dari modul-modul *input* dan modul-modul *output* dengan menggunakan tegangan DC24volt sebagai sumbernya. Untuk siemens S7-300 ini, modul-modul *input* dan *output digital* yang digunakan adalah dengan format modul 32 bit. Jadi misalnya untuk slot pertama untuk I/O (slot 4) akan terdapat alamat I/O digital sebagai berikut :

Tabel 2.2 Urutan jumlah bit input dalam satu slot modul.

I0.0	I1.0	I2.0	I3.0
I0.1	I1.1	I2.1	I3.1
I0.2	I1.2	I2.2	I3.2
I0.3	I1.3	I2.3	I3.3
I0.4	I1.4	I2.4	I3.4
I0.5	I1.5	I2.5	I3.5
I0.6	I1.6	I2.6	I3.6
I0.7	I1.7	I2.7	I3.7

Dan begitu juga untuk alamat output-nya, yaitu 32 bit keluaran. Namun karena terletak pada slot ke-5 maka alamat harus mengikuti dari alamat slot sebelumnya, yaitu melanjutkan dari slot ke-4, diawali dengan Q4 s/d Q7 dengan masing-masing memiliki 8bit keluaran. Perangkat I/O PLC kali ini tidak menggunakan fasilitas modul analog, karena seluruh sinyal input dan output PLC adalah berupa digital.

### 3. Interface modul

*Interface modul* atau yang biasa disingkat dengan IM ini adalah modul yang digunakan untuk menghubungkan antara satu *rack* dengan *rack* lainnya. Atau dengan kata lain untuk menghubungkan *central rack* dengan *expansion rack* dibutuhkanlah IM. Hal ini dapat terjadi apabila banyak I/O modul yang dibutuhkan dan tidak mencukupi untuk disusun dalam satu *rack* saja, jadi harus ada penambahan *rack* untuk penempatan modul-modul I/O tersebut.

### 4. Ethernet TCP/IP

Modul ini berfungsi sebagai sarana konektifitas *multi access* pada PLC, pertukaran data dengan PC yang mempunyai level lebih tinggi. Jadi modul ini memungkinkan komputer-komputer lain dalam satu jaringan (yang diberikan akses) untuk dapat *on-line* dengan PLC. Namun tetap komputer-komputer tersebut tidak dapat mengubah atau memodifikasi program-program yang sudah ada di dalam PLC tersebut. Ethernet ini sangat diperlukan program SCADA yang akan digunakan sebagai interface pada PLC.

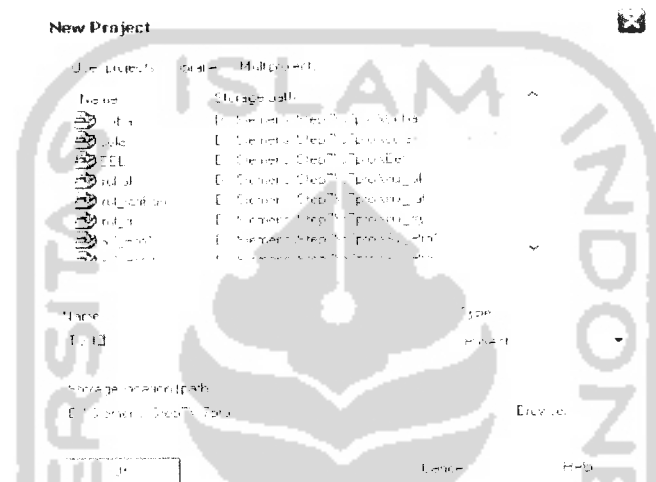
#### 2.3.4 Konfigurasi hardware

Sebelum dioperasikan, PLC haruslah di konfigurasi terlebih dahulu. Proses ini dilakukan pada PC pada beberapa perangkat lunak. Proses ini juga disebut sebagai proses *interfacing* dari PC ke PLC. *Interfacing* ke PLC Siemens S7-200 dan S7-300 membutuhkan prosedur konfigurasi pada dua tempat. Konfigurasi pertama dilakukan di *software Simatic STEP7* (Siemens), konfigurasi

ini ditujukan untuk modul *Ethernet TCP / IP*. Dilanjutkan dengan konfigurasi kedua yang dilakukan di *software WinCC*.

### 2.3.5 *Software simatic step 7*

1. Masukkan new project pada simatic step 7.



Gambar 2.4 new project pada simatic step 7

2. Masukkan *simatic station 300* pada step 7

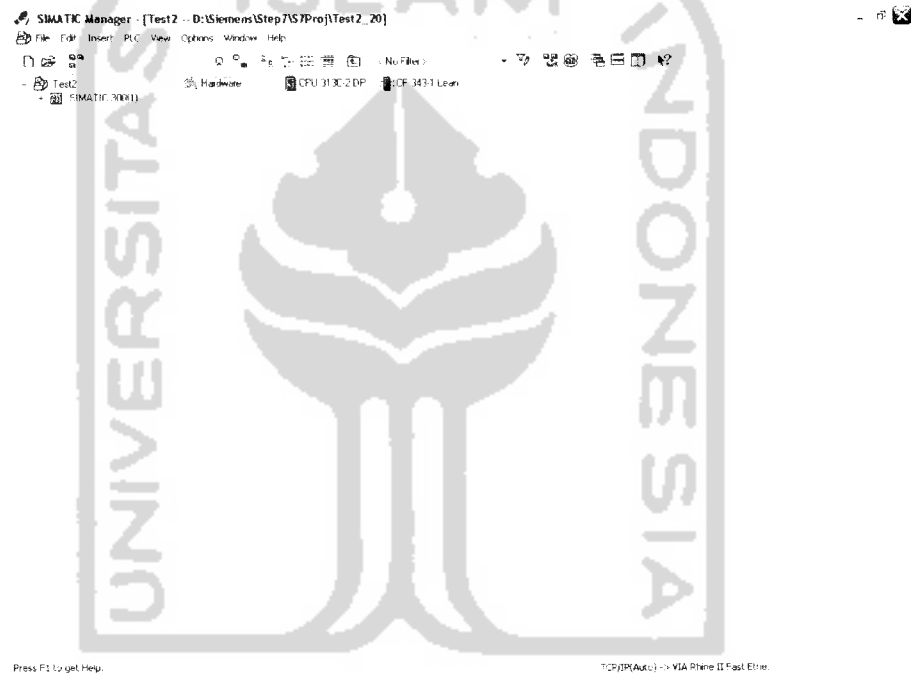


Inserts SIMATIC 300 Station at the cursor position.

Gambar 2.5 insert simatic station 300



3. Pada tampilan *simatic manager step 7*, klik 2 kali pada *hardware CPU 313C-2DP*. Maka akan muncul *hardware configuration*, dibagian ini terdapat tabel kosong dengan urutan nomer yang terurut 1 – 11, tabel ini harus di isi dengan modul-modul yang digunakan sesuai dengan urutan slot dalam satu rack (1 – 11). Lalu masukkan CP 343-1, yaitu *modul Ethernet untuk S7-300*.



Gambar 2.6 tampilan simatic manager step 7

HW Config [SIMATIC 300(1) (Configuration) Test2]

Station Edit Insert PLC View Options Window Help

Find

Profile Standard

- PROFIBUS DP
- PROFIBUS PA
- PROFINET IO
  - AI-Maximal Field Devices
  - Gateway
  - IO
- SIMATIC 300
  - C7
  - CP 300
  - CPU 300
  - FM 300
  - Gateway
  - IM 300
  - MP-EXTENSION
  - PS 300
  - RACK-300
  - SM 300
- SIMATIC 400
  - SIMATIC IFC Based Control
  - SIMATIC PC Station

PROFIBUS(2) DP-master system(1)

[3] EM 277

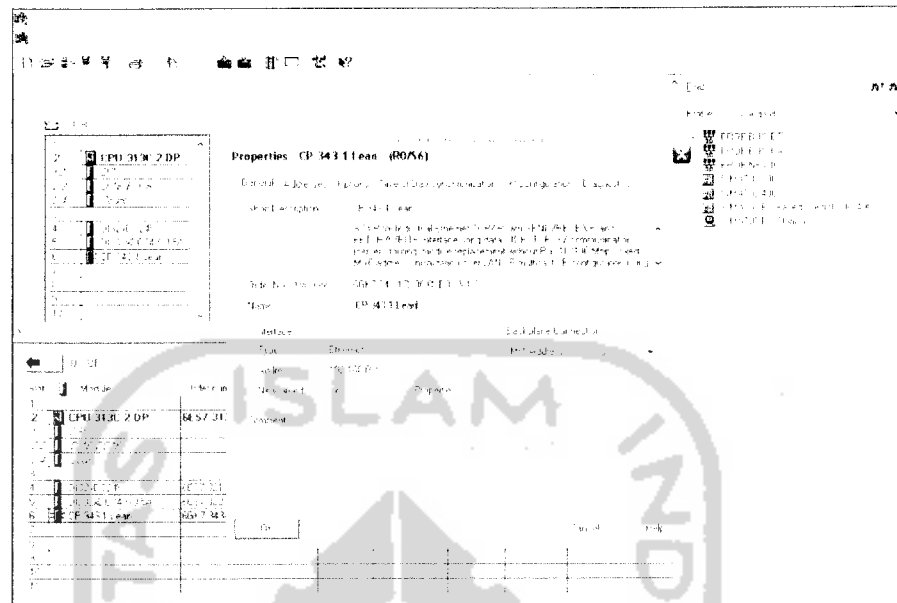
[IP-NORM]

01 UR

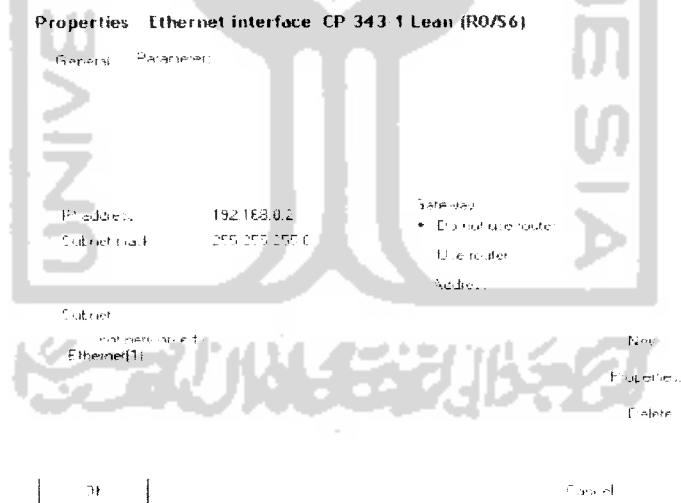
Slot	Module	Order number	Firmware	MPI address	I address	Q address	Comment
1							
2	CPU 313C-2 DP	6ES7 313-6CE01-0AB0	V2.0	5			
2.2	DP				124		
2.2	DI16-DO16				124	124	
2.4	Count				768	768	
3							
4	CP 343-1 Leon	6GK7 343-1EX30-0XE0	V1.0	6	272	287	
5							

Gambar 2.7 Konfigurasi hardware S7-300

- Klik dua kali di tulisan CP 343-1, maka akan muncul *window* berikut. Perhatikan pada bagian *Networked* masih berstatus “No”.
- Untuk mengaktifkan status *network*, klik *icon properties*, lalu kemudian isikan lah *IP address* dan *subnet mask*. Lalu klik *icon New* dan pilih *Ethernet(1)*.



Gambar 2.8 Konfigurasi ethernet 1.



Gambar 2.9 Properties ethernet interface CP 343-1 lean.

6. *Networked* akan berubah statusnya menjadi *Yes* setelah melakukan prosedur diatas.

7. *Download Hardware Configuration* di atas, kemudian *start* PLC.

Pastikan bahwa tidak ada lampu merah yang menyala di CPU dan CP 343-1.

Pada susunan konfigurasi diatas, urutan pertama di isi oleh *power supply* dari PLC. Pada urutan ke-dua di isi dengan CPU dari PLC, setelah memasukkan **CPU 313C-2 DP** sebagai tipe dari CPU yang digunakan, maka akan keluar sub slot lagi dari slot dua (2), yaitu DP, DI16/DO16, dan Counter. DP atau disebut juga '*Distributed Pheripheral*' berfungsi sebagai komunikasi untuk *PROcess Field BUS* (PROFIBUS). CPU ini juga telah memiliki 16bit Digital Input dan 16bit *Digital Output*. Sedangkan *Counter* adalah fungsi pencacah. Masing-masing *input*, *output*, maupun *counter* telah memiliki alamatnya masing-masing. Namun pada pemrograman PLC untuk *mini plant* ini tidak menggunakan I/O dari CPU, tapi menggunakan I/O tambahan yaitu menggunakan modul I/O tambahan pada slot ke-4 dan ke-5. Hal ini dikarenakan modul I/O tambahan ini memiliki lebih banyak jumlah bit-nya dan memiliki alamat yang lebih mudah yakni dengan awalan I/O dari 0 s/d 3 dengan masing-masing 8bit, jika dibandingkan dengan I/O dari CPU dengan awalan 124 dengan jumlah 16bit.

#### 2.4 SCADA (Supervisory Control And Data Acquisition)

SCADA merupakan sistem elektronik yang digunakan untuk memonitor dan mengontrol suatu perangkat pemrosesan. Pada sektor energi, SCADA digunakan untuk mengontrol transmisi dan distribusi arus listrik, mengatur buka/tutup keran minyak dan gas pada pipa-pipa dan *reservoir* , mengatur debit air dari waduk untuk keperluan pembangkit tenaga listrik. SCADA juga

digunakan oleh sektor lainnya untuk keperluan, misalnya mengontrol dan memonitor pemrosesan bahan-bahan kimia, proses sanitasi air, dan sebagainya.

Menurut laporan US Department of Energy bulan July 2003, banyak sistem SCADA yang dirancang tanpa atau sedikit sekali memperhatikan faktor keamanan. Sebagai contoh, data yang dikirim melalui sistem SCADA dilakukan melewati jaringan kabel publik ataupun jaringan nirkabel tanpa enkripsi. Protokol yang digunakan untuk menerima perintahpun dapat diakses tanpa autentikasi.

Selain itu, beberapa sistem SCADA menggunakan sistem operasi yang tidak aman. Kerentanan-kerentanan di atas jelas-jelas memberi peluang besar bagi orang untuk melakukan eksploitasi terhadap sistem SCADA. Di samping itu, karena sistem SCADA sifatnya *real-time*, maka ketersediaan sistem tersebut menjadi sangat penting. Bayangkan jika perintah membuka saluran gas untuk mengurangi tekanan pada *reservoir* tidak sampai ke sistem SCADA, karena jalur data menjadi penuh akibat *denial of service attack*.

Suatu proses yang dijalankan dalam sebuah industri kini semakin berkembang dan semakin rumit, karena seiring dengan semakin tingginya kualitas produk yang akan dihasilkan, semakin banyak pula mesin-mesin yang digunakan, semakin tingginya tingkat efisiensi dan keamanan yang harus dimaksimalkan dan otomatis semakin banyak pula proses-proses yang dilakukan.

Proses-proses tersebut haruslah bekerja dibawah pengawasan dari operator, mereka juga dituntut untuk mempunyai akses yang cepat untuk mendapatkan informasi dan mengetahui kondisi sistem dari waktu ke waktu

dalam menjalankan sebuah sistem kontrol kendali tingkat tinggi, hal inilah yang telah membuktikan betapa pentingnya sebuah teknologi yang disebut dengan *Human Machine Interface* (HMI).

*Human machine interface* berarti mengontrol lajunya suatu proses dan tetap menjaga agar suatu sistem dapat bekerja dengan baik pada level maksimumnya. SCADA sebagai antar muka yang dapat memudahkan proses pengontrolan, yang berfungsi untuk mengoperasikan dan memonitor jalannya suatu proses atau sistem yang mana sebelumnya telah dikontrol oleh PLC. SCADA memberikan wujud visualisasi dari sistem yang telah dibangun melalui serangkaian bentuk *display animasi mimic diagram* yang dirancang hingga menyerupai kondisi sebenarnya di lapangan.

Jadi dengan menggunakan perangkat SCADA ini operator tidak hanya dapat memantau seluruh aktifitas dan kondisi mesin, tapi juga dapat berperan sebagaimana layaknya peranan operator yaitu mengoperasikan dan mengontrol sistem tersebut dengan hanya melihat pada layar komputer saja, seperti aktifitas pembacaan sensor - sensor, mengumpulkan laporan - laporan yang berkenaan dengan proses produksi, memastikan proses *sequential*, memberikan peringatan keselamatan saat mesin akan dijalankan, dapat langsung mengidentifikasi *error* atau kerusakan pada sistem, dan juga dapat melakukan *emergency stop* apabila terjadi suatu kondisi darurat, dan lain sebagainya.

Karena PLC yang digunakan adalah PLC Siemens, maka sebenarnya Siemens sendiri juga menyediakan *SIMATIC panel PCs* yang berfungsi sebagai HMI-SCADA khusus untuk PLC-PLC Siemens. *Simatic panel PCs* ini berbentuk

seperti *console* yang dilengkapi dengan layar *monitor* untuk *display operasional system*, jadi sifatnya *portable*.

SCADA adalah suatu sistem pengendalian alat secara jarak jauh, dengan kemampuan memantau data-data dari alat yang dikendalikan. SCADA merupakan bidang yang secara kontinyu selalu dikembangkan di seluruh bagian dunia pada berbagai tipe industri yang menghabiskan bertrilyun-trilyun rupiah.

Era komputerisasi melaju dengan cepat. Tak pelak lagi, data dan informasi bergerak dengan cepat dari satu tempat ke tempat lain. Diperlukan penanganan yang tepat dan benar akan data dan informasi tersebut agar dapat dikelola dengan baik dan hasilnya dapat memberikan manfaat yang optimal bagi pengguna. Penanganan ini sering disebut sebagai *database*.

Di dalam sistem SCADA, database sebagai media untuk pengelolaan data-data real-time menjadi tulang punggung bagi para tenaga ahli SCADA dalam melakukan analisa, studi, pengembangan serta memberikan output bagi operasi sistem yang dibuat. Dengan adanya fasilitas *sampler* pada sistem *software* SCADA, pemberdayaan database ini bahkan lebih mudah, gampang dan efektif. Hasilnya bisa diakses oleh para tenaga ahli dan datanya digunakan untuk keperluan khusus maupun hanya sebagai tampilan saja. Definisi database dari *sampler* ini bisa diakses melalui saluran komunikasi *TCP/IP*.

Penanganan gangguan juga lebih optimal serta waktu pemulihannya lebih cepat karena dilakukan secara terkoordinir serta satu komando. Secara garis besar, sistem SCADA meliputi:

- a. Akuisisi data.

- b. Supervisory control.
- c. Pemantauan data, pemrosesan event (kejadian) dan alarm.
- d. Kalkulasi data.
- e. Tagging (penandaan).
- f. Perekaman data.
- g. Pelaporan.

## 2.5 Pemrograman SCADA dengan WinCC

Sesuai dengan namanya yaitu “pengawasan kontrol dan pengumpulan data”, sangat jelas tergambar fungsi dari SCADA itu sendiri, yaitu melakukan pengawasan dan mengambil semua informasi dari sistem yang di kontrol. SCADA bertindak sebagai pengawas dan memberikan perintah atas semua yang telah dikontrol oleh PLC.

Pemrograman yang dikerjakan pada SCADA dilakukan dengan menggunakan *Software WinCC* dari *siemens*. Hampir secara keseluruhan dari pemrograman yang dilakukan adalah berhubungan dengan animasi. Animasi dapat menyatakan kondisi sistem yang sedang dikontrol, seperti kondisi berjalan, kondisi *ready*, kondisi *fault*, dan sebagainya.

WinCC (*Windows Control Centre*) dari *simatic siemens* ini memiliki kekurangan dan kelebihan tersendiri, antara lain :

Kekurangannya :

- a. Butuh RAM (*Random Access Memory*) yang lebih tinggi dari komputer biasanya, contoh minimal RAM yang digunakan yaitu 256 MB.



- b. Bagi kalangan pemula, penggunaan WinCC lebih sulit untuk dipahami dibandingkan dengan *InTouch* dari *Wonderware*.

Kelebihannya :

- a. Mendukung teknologi profibus.
- b. Sangat baik dalam penggunaan satu platform, seperti semua hardware (PLC) berasal dari *vendor* yang sama.



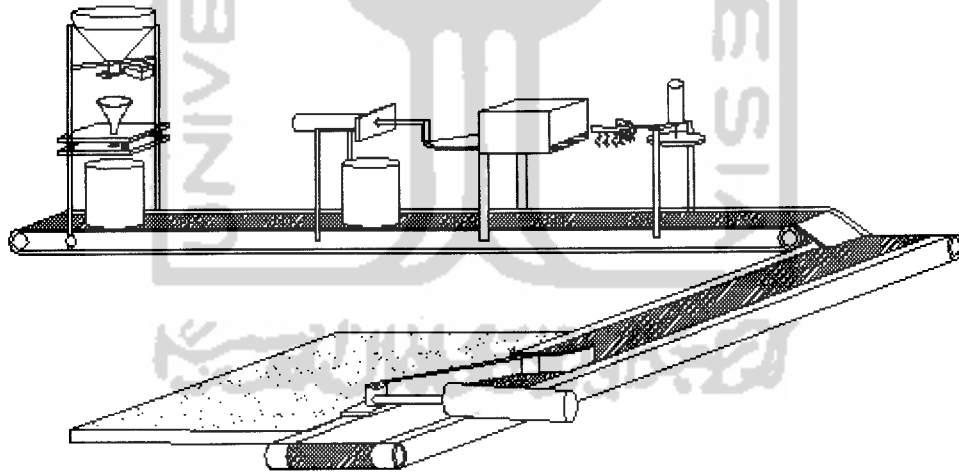
## BAB III

### PERANCANGAN SISTEM

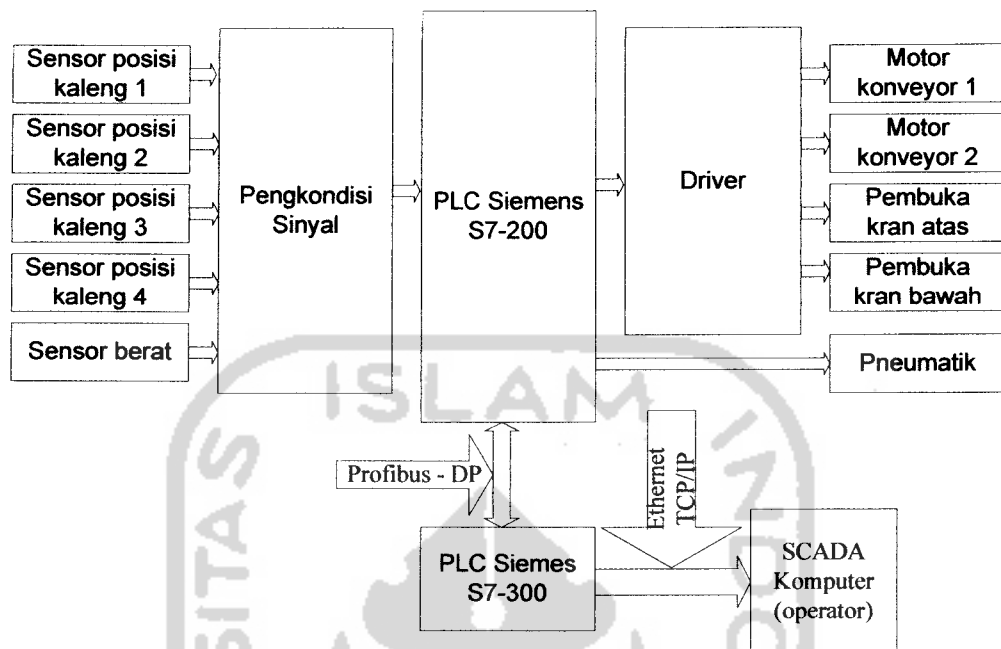
#### 3.1 Perancangan perangkat keras

Pada perencanaan sistem tersebut ada tiga bagian yang perlu mendapat perhatian yaitu:

- a. Pengisian dan penutupan kaleng susu secara otomatis yang difungsikan sebagai *plant*.
- b. PLC yang merupakan “otak” dari system.
- c. *SCADA software (WinCC)* yang akan *memvisualisasikan* proses yang terjadi pada *plant*.



Gambar 3.1 *Plant* yang digunakan



Gambar 3.2 Blok diagram perancangan sistem

PLC siemens S7-200 menerima masukan berupa sensor dan saklar. Masukan-masukan diolah oleh CPU menghasilkan keluaran yang dapat menggerakkan mekanisme *output device*. Dari PLC siemens S7-200 dikomunikasikan dengan PLC siemens S7-300 menggunakan sistem profibus. Sedangkan antarmuka antara komputer operator dengan *device* menggunakan Ethernet TCP/IP.

Perangkat keras yang digunakan berupa sensor *infrared*, pneumatik, *motor driver*, motor AC satu fasa 1420 rpm dan satu unit PLC siemens S7-300 lengkap dengan komunikasi protokolnya.

### 3.1.1 Motor Driver

Rangkaian ini dibangun dengan komponen utama berupa relay AC. Keluaran PLC akan membuat relay aktif, keluaran relay dihubungkan ke motor.

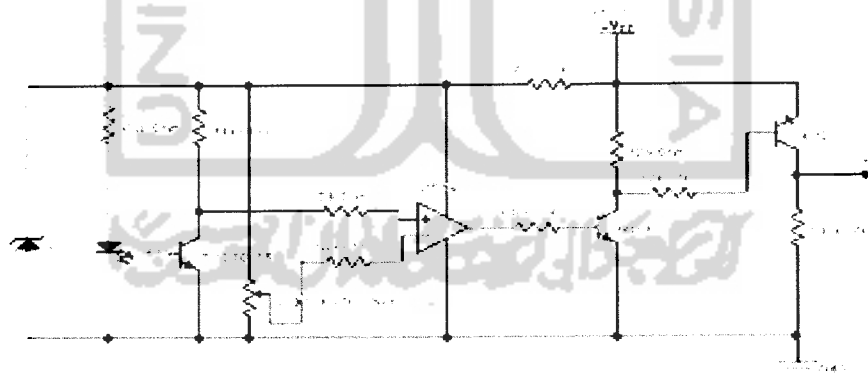
Gambar 3.3



Gambar 3.3 Motor driver

### 3.1.2 Sensor

Pada sistem ini digunakan sensor *infrared* yang berfungsi untuk mendeteksi ada tidaknya kaleng.



Gambar 3.4 Sensor pendeteksi kaleng

Pada penelitian ini digunakan 4 buah sensor *infrared* yang bekerja secara terpisah tetapi mempunyai fungsi sama yaitu untuk mendeteksi kaleng. Perbedaan keempat sensor tersebut hanya terletak pada tegangan referensi ( $V_{ref}$ ) pada masukan in (-)

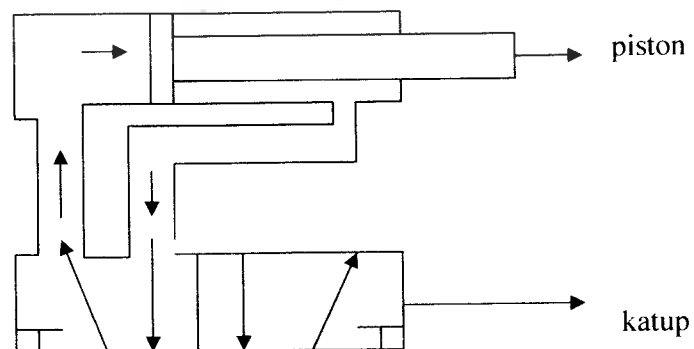
sensor tersebut hanya terletak pada tegangan referensi ( $V_{ref}$ ) pada masukan in (-) pada komparator. Tegangan referensi dapat diatur dengan resistor variabel yang berfungsi untuk mengatur kesensitifitasan sensor *infrared*.

### 3.1.3 Sistem Pengisian

Sistem ini digunakan untuk mengisi kaleng dengan bubuk. Sistem ini terdiri dari 2 tangki, yaitu tangki besar dan tangki kecil. Tanki besar berfungsi untuk mengisi tangki kecil yang berfungsi sebagai takaran berat pasir yang diisikan ke kaleng. Mekanisme pembuka dan penutup kran tangki menggunakan kontaktor magnetis sebagai penarik kran.

### 3.1.4 Sistem Pneumatik

Sistem pneumatik digunakan untuk memindahkan beban dengan mengontrol aliran tekanan udara dalam piston dan jalur distribusinya menggunakan katup elektromekanik. Contoh sistem pneumatik dapat dilihat pada Gambar 3.5. Sistem yang digambarkan adalah tabung dalam kondisi keluar, kondisi tabung dikontrol menggunakan katup.



Gambar 3.5 Ilustrasi sistem pneumatik

### 3.1.5 Konfigurasi Dua PLC

Mengatur dan menyusun pola kerja menjadi dua bagian yaitu :

- a. *Mengkonfigurasi* antara dua *PLC* yaitu *PLC* siemens S7-200 dan S7-300.
- b. *Mengkonfigurasi* antara *PC (software)* dengan *device (PLC)*.

*Konfigurasi* antara dua *PLC* telah dilakukan dan hasilnya sukses. Dimana I/O S7-200 dapat dibaca di S7-300 dalam pengertian *PLC* siemens S7-200 dapat berkomunikasi dengan *PLC* siemens S7-300. Adapun *interface* antara dua *PLC* tersebut menggunakan sistem *profibus (Process fieldbus)*. Untuk sistem ini menggunakan beberapa komponen seperti :

- a. *PLC* siemens S7-200 CPU 224 AC/DC/RLY (214-1BP22-0XB0).
- b. EM 277-Profibus DP (6ES7 277-0AA22-0XA0).
- c. *PLC* siemens S7-300 CPU 313C-2DP (313-6CE01-0AB0).
- d. CP 343-1 lean simatic net (6GK7 343-1CX00-0XE0).
- e. *PC adapter USB* (6ES7972-0CB20-0XA0) SVPT4.
- f. Kabel profibus.

Langkah-langkah konfigurasi antara dua *PLC*, antara lain yaitu :

- 1 Hal pertama yang dilakukan adalah menghubungkan EM 277 ke *PLC* S7-200 CPU 224 AC/DC/RLY..
2. Mendownload program menggunakan *Software Microwin* ke *PLC* siemens S7-200 sampai sukses.
3. Menghubungkan CP 343-1 lean ke *PLC* siemens S7-300 CPU 313C-2DP.

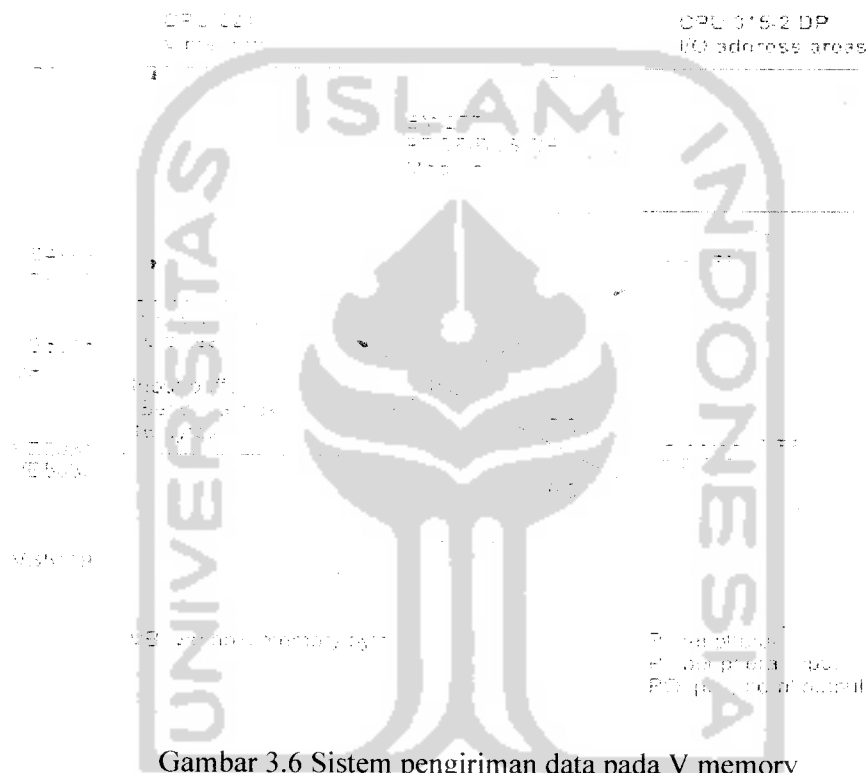
4. Membuat new project di software *Step7 (Simatic Manager)*.
5. Insert new object -> "simatic 300 station".
6. Pada "*HW Config*" dengan cara double klik, lalu masukkan :
  - a. Rack-300
  - b. CPU 313C-2DP, pilih sesuai order no yang ada di depan module CPU (6ES7....)
  - c. CP 343-1 Lean, set IP address.
  - d. EM 277 profibus-DP.
  - e. *Compile* dan *download* hingga sukses.
  - f. Sambungkan EM 277 ke CPU313C-2DP menggunakan kabel Profibus.
7. *Run* semua *PLC* (dengan catatan semua indicator harus menyala hijau).

### 3.2 *Profibus*

*Profibus DP (Distributed Peripheral)* adalah dirancang untuk komunikasi data kecepatan tinggi pada level alat. Protokol dan fungsi dirancang berdasarkan standar Eropa EN 50 170. Pengontrol *profibus* (PLC) berkomunikasi dengan *plant* ataupun objek alat yang dikendalikan dengan kecepatan tinggi. *Profibus* mampu mengembangkan analisa yang baik dan jaringan sederhana.

*EM 277 profibus-DP* mendukung beberapa modul I/O (input dan output) untuk konfigurasi ke aplikasi yang lebih luas. Perbedaan ukuran input dan output pada pengiriman data antara master dengan *EM 277* memerlukan aplikasi yang lebih spesifik. *EM 277* menulis output menuju area *V memory* di dalam *CPU*

(*Central Processing Unit*). Transmisi data antara master dan DP slave yaitu terbagi kedalam 3 fase : *Parameter Assignment* (ukuran penempatan), Konfigurasi, dan *Data Exchange* (pertukaran data). Dua fase pertama secara normal hanya terdapat sekali bila sistem dimulai atau bila slave ditambah.



Gambar 3.6 Sistem pengiriman data pada V memory

### 3.3 Diagram alir sistem

Pengendalian keseluruhan sistem merupakan kendali berurutan, dimana setiap keadaan mempengaruhi keadaan yang lain. Urutan pengendalian sistem dapat digambarkan dengan diagram alir, sebagaimana dapat dilihat pada Gambar 3.7 dan 3.8.



### 3.3.1 Diagram Alir Sistem Pada Konveyor 1



Gambar 3.7 Diagram alir sistem pada konveyor 1

### 3.3.2 Diagram Alir Sistem Pada Konveyor 2



Gambar 3.8 Diagram alir sistem pada konveyor 2

Dari Gambar 3.7 dan 3.8 diagram alir sistem dapat dijelaskan bahwa ketika sistem ON maka perintah menghidupkan motor 1 dan motor 2 akan bekerja. Hasil perputaran motor 1 akan menggerakkan konveyor 1 dan hasil pergerakan motor 2 akan menggerakkan konveyor 2. Pergerakan *belt* konveyor ini digunakan sebagai media perpindahan kaleng.

Pada konveyor 1 saat sensor 1 mendeteksi kaleng, maka motor akan berhenti, sehingga proses dilanjutkan dengan pengisian. Kemudian motor akan

berputar kembali sampai kaleng mencapai sensor 2. Maka motor akan berhenti lagi, proses dilanjutkan dengan menempatkan tutup kaleng pada mulut kaleng oleh mekanisme pneumatik. Kemudian motor akan berputar kembali sampai pada sensor 3, maka motor akan berhenti lagi, lalu proses akan dilanjutkan dengan menekan tutup kaleng oleh mekanisme pneumatik agar tutup menjadi lebih rapat. Selanjutnya kaleng akan menuju konveyor 2 dan berpindah ke konveyor 2.

Pada konveyor 2 saat sensor mendeteksi adanya kaleng, maka kaleng akan didorong ke pinggir konveyor oleh mekanisme pneumatik. Setelah kaleng berada di tempatnya proses akan kembali seperti semula.

### 3.4 Alamat PLC yang digunakan

Langkah pertama dalam pembuatan *ladder* dan *mapping* PLC adalah menentukan alamat masukan dan keluaran. Alamat yang digunakan pada pembuatan *ladder* sistem dapat dilihat pada tabel 3.1.

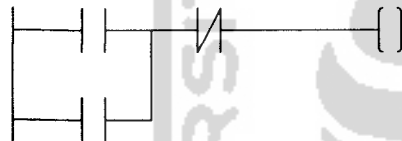
Tabel 3.1 Alamat PLC yang digunakan

WinCC	M S7-300	S7-300	S7-200	I/O S7-200	Fungsi
M 10.0	M 10.0	I 256.0	V5016.0	I 0.0	Start
M 10.1	M 10.1	I 256.1	V5016.1	I 0.1	Stop
Q 4.2	M 10.2	I 256.2	V5016.2	I 0.2	Sensor pendeteksi kaleng posisi 1
Q 4.4	M 10.4	I 256.4	V5016.4	I 0.4	Sensor pendeteksi kaleng posisi 2
Q 4.5	M 10.5	I 256.5	V5016.5	I 0.5	Sensor pendeteksi kaleng posisi 3
Q 4.7	M 10.7	I 256.7	V5016.7	I 0.7	Sensor pendeteksi kaleng 4
Q 5.0	M 11.0	I 257.0	V5017.0	I 1.0	Sensor berat
Q 4.2	M 10.2	I 256.2	V5016.2	Q 0.0	Pembuka kran tangki kecil
Q 4.1	M 12.1	I 258.1	V5018.1	Q 0.1	Pembuka kran tangki besar
Q 6.2	M 12.2	I 258.2	V5018.2	Q 0.2	Motor konveyor 1
Q 6.2	M 12.3	I 258.3	V5018.3	Q 0.3	Motor konveyor 2

Lanjutan Tabel 3.1 Alamat PLC yang digunakan

WinCC	M S7-300	I S7-300	Q S7-200	I/O S7-200	Fungsi
Q 4.5	M 12.7	I 258.7	V5018.7	Q 0.7	Piston 1 ( penekan tutup kaleng )
Q 4.7	M 13.0	I 259.0	V5019.0	Q1.0	Piston 2 ( pendorong kaleng )
Q 4.4	M 13.1	I 259.1	V5019.1	Q1.1	Piston 3 ( penempatan tutup kaleng )

### 3.4.1 Diagram *ladder* untuk *start* dan *stop* sistem

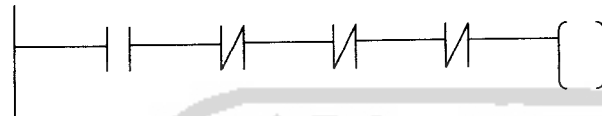
Gambar 3.9 Diagram *ladder* untuk *start* dan *stop* system

Pembuatan *ladder* untuk *start* dan *stop* berfungsi untuk memastikan agar sistem pasti pada kondisi ON pada saat tombol *start* ditekan dan pada kondisi OFF pada saat tombol *stop* ditekan. M0.0 merupakan alamat internal relay yang digunakan sebagai keluaran yang tidak berhubungan langsung dengan modul keluaran. Keluaran M0.0 digunakan pada setiap awal pembuatan diagram *ladder*.

### 3.4.2 Diagram *ladder* pada konveyor 1

Pada konveyor 1 terdapat tiga subsistem yaitu pengisian kaleng, penempatan tutup kaleng, dan penekan tutup kaleng. *Ladder* utama pada konveyor 1 merupakan gabungan keluaran dari masing-masing subsistem. Gabungan dari

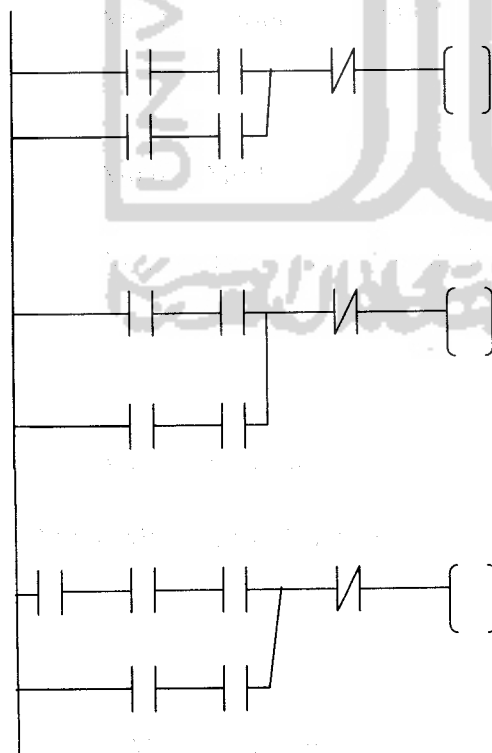
keluaran-keluaran subsistem dihubungkan secara seri sehingga kalau ada subsistem yang belum selesai menjalankan prosesnya motor tidak akan berputar.

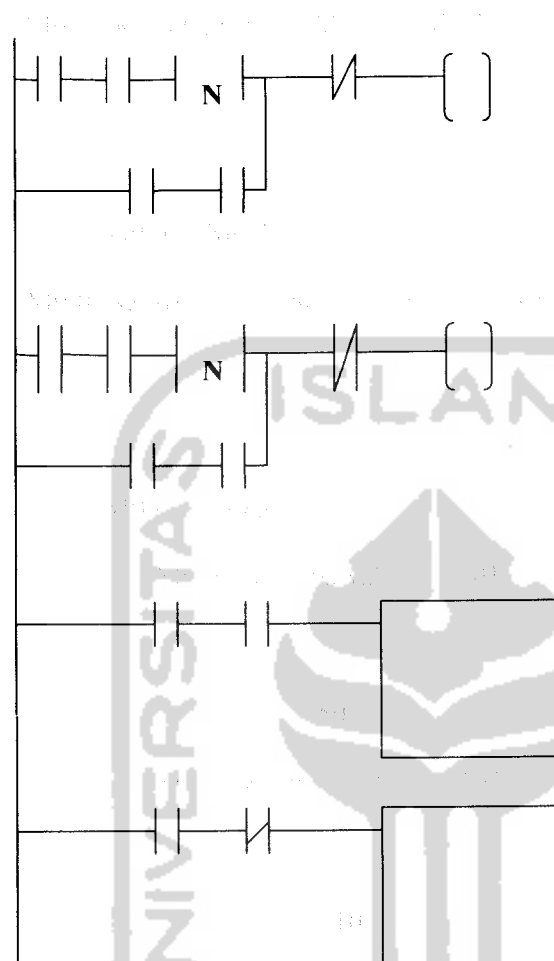


Gambar 3.10 Diagram *ladder* utama pada konveyor 1

#### 3.4.2.1 Diagram *ladder* pada pengisian kaleng

Pengisian kaleng terjadi pada saat sensor posisi kaleng 1 mendeteksi adanya kaleng dan jika berat bubuk pada takaran sudah terpenuhi. Jika salah satu kondisi tersebut tidak terpenuhi maka proses pengisian tidak akan terjadi. Dengan kondisi tersebut maka didapatkan diagram ladder seperti pada Gambar 3.10

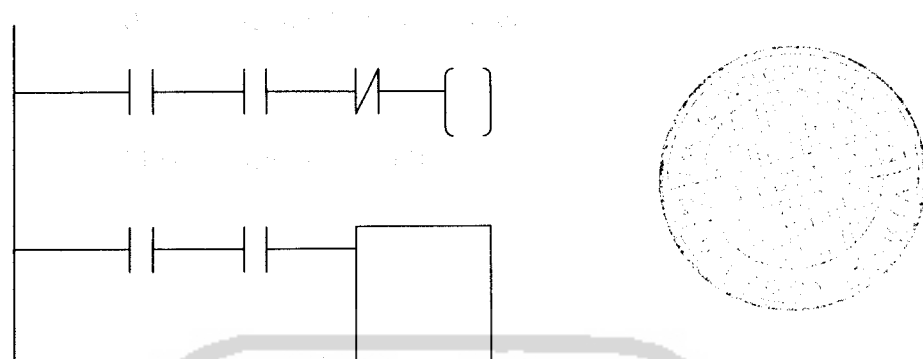




Gambar 3.11 Diagram *ladder* pada pengisian kaleng

#### 3.4.2.2 Diagram *ladder* untuk penempatan tutup kaleng

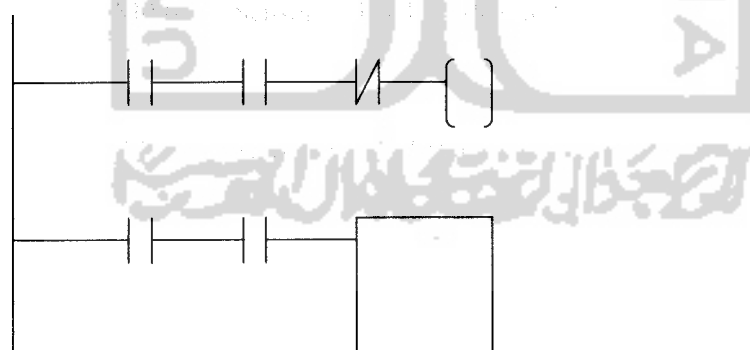
Proses penempatan tutup kaleng terjadi pada saat sensor pendeteksi kaleng posisi 2 mendeteksi adanya kaleng. Sensor akan memberi masukan pada PLC sehingga keluaran yang terhubung pada katup akan aktif dan menggerakkan piston 1. Dorongan piston 1 akan menempatkan tutup kaleng ke mulut kaleng. Piston kembali ke posisi semula setelah 2 detik.



Gambar 3.12 Diagram *ladder* untuk penempatan tutup kaleng

### 3.4.2.3 Diagram *ladder* penekan tutup kaleng

Proses penekanan tutup kaleng terjadi pada saat sensor pendeteksi kaleng posisi 3 mendeteksi adanya kaleng. Sensor akan memberi masukan pada PLC sehingga keluaran yang terhubung pada katup akan aktif dan menggerakkan piston 2. Dorongan piston 2 akan menekan tutup kaleng. Piston kembali ke posisi semula setelah 2 detik.

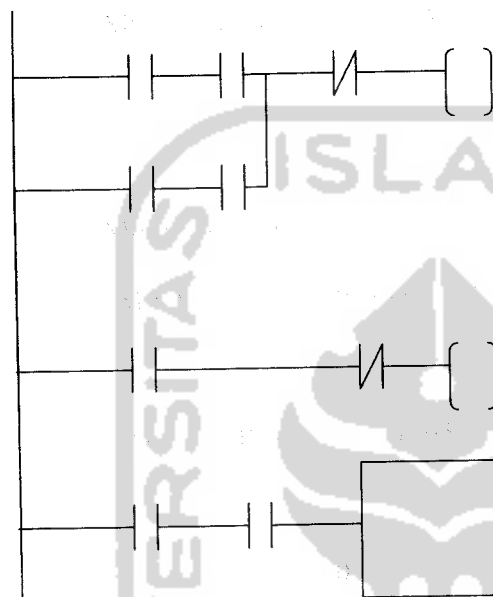


Gambar 3.13 Diagram *ladder* penekan tutup kaleng

### 3.4.3 Diagram *ladder* pada konveyor 2

Proses penempatan kaleng terjadi pada saat sensor pendeteksi kaleng posisi 4 mendeteksi adanya kaleng. Sensor akan memberi masukan pada PLC

sehingga keluaran yang terhubung pada katup akan aktif dan menggerakkan piston 3. Dorongan piston 2 akan menggerakkan mekanisme penempatan kaleng ke pinggir konveyor 2. Piston kembali ke posisi semula setelah 2 detik.



Gambar 3.14 Diagram *ladder* pada konveyor 2

### 3.5 Perancangan Perangkat Lunak

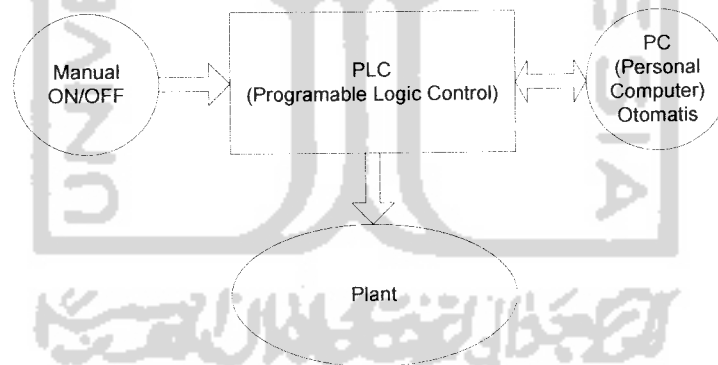
Perancangan program dimulai dengan pembuatan program anak tangga dengan menggunakan diagram ladder pada perangkat lunak *Microwin* pada PLC Siemens S7-200 hanya saja pada *plant* yang digunakan telah diprogram sebelumnya oleh *programmer* sedangkan *SIMATIC S7* untuk memprogram PLC Siemens S7-300 dan mengkonfigurasi dengan PLC Siemens S7-200, sehingga dapat *meng-inisialisasikan* masukan – masukan ke PLC dan juga keluarannya. Karena pemrograman *SCADA* merupakan penggabungan antara pemrograman PLC dan PC, pemrograman juga dilakukan pada satu lagi perangkat lunak yaitu



*WinCC*, yaitu untuk merancang program animasi *mimic diagram* dan juga pengalamatan perintah sesuai dengan *plant* yang digunakan.

### 3.6 Jenis kontrol yang digunakan

Sistem kontrol pengisian dan penutupan kaleng pada penelitian ini bersifat aktif dari dua jalur, yaitu dari PC ke PLC dan dari manual ke PLC. Pengontrolan secara manual disini tidak dihapuskan, karena pada kenyataan di lapangan kontrol ini masih sangat diperlukan. Satu unit *plant* memiliki dua buah saklar *push button*, yaitu satu untuk mengaktifkan dan lainnya untuk mematikan. Oleh karena itu sistem ini juga ditujukan untuk menggabungkan kedua sistem pengontrolan diatas, dalam artian bila pengontrolan dilakukan pada satu sisi maka secara otomatis juga akan mengaktifkan pengontrolan pada sisi lainnya.



Gambar 3.15 Dua macam pengontrolan yang digunakan

Sedangkan pengontrolan secara otomatis pada PC yang berbasis SCADA diaplikasikan menggunakan perangkat lunak yang bernama **WinCC** dari **siemens**. Proses pemrograman SCADA berupa pembuatan animasi, pengalamatan *address* dan *tag name*, serta penulisan *script* dikerjakan pada perangkat lunak WinCC.

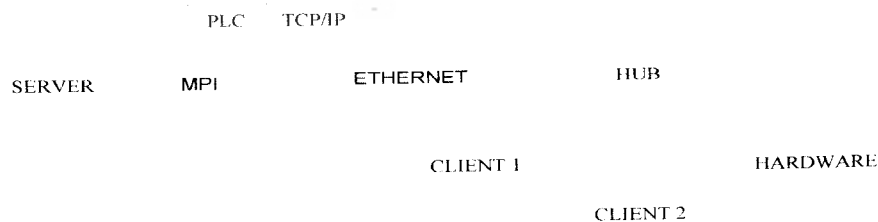
Setelah melakukan pembacaan kasus dan merancang pemrograman PLC, *programmer* tentunya sudah memiliki seluruh alamat *input / output* yang digunakan dalam pemrograman tersebut, dan tentunya juga telah mengetahui deskripsi apa saja yang diwakili oleh alamat-alamat tersebut. Karena pada program SCADA ini juga akan memanfaatkan alamat-alamat I/O yang telah dipergunakan sebelumnya pada pemrograman PLC, jadi dengan memanfaatkan alamat-alamat I/O tersebut SCADA akan dapat mendeteksi *input* maupun *output* yang sedang beroperasi dan menampilkannya kedalam bentuk animasi yang akan merepresentasikan keadaan sebenarnya di lapangan.

Pada PLC juga harus dilengkapi dengan *modul* TCP / IP untuk *Ethernet*. Karena SCADA hanya dapat meng-akses PLC melalui *ethernet*, maka konfigurasi modul ini pada saat pemrograman PLC.

### 3.7 *Supervisory Control And Data Acquisition (SCADA)*

#### 3.7.1 Prinsip kerja SCADA

Pada dasarnya *SCADA* adalah sebuah sistim komunikasi kontrol data yang berbasiskan pada jaringan komputer.



Gambar 3.16 Penggunaan PLC bersama

Dari Gambar diatas dapat dilihat bahwa tidak hanya *server* saja yang dapat meng-akses *PLC*, tetapi komputer-komputer lain (yang diberikan akses) juga dapat untuk *on-line* dengan *PLC*. Dengan adanya fasilitas *ethernet* yang disediakan oleh *PLC*, maka *client-client* ini dapat mengambil dan meng-akses data dari *PLC*, serta dapat juga merubah-rubah program yang ada didalam *PLC*. Tetapi tentunya tidak lupa sebelumnya haruslah ditetapkan *IP Address* dari *PLC* yang akan diakses (di dalam suatu jaringan, *PLC* dianggap sebagai sebuah *PC* yang juga harus mempunyai alamat *IP*) agar *PLC* tersebut dapat dibaca sebagai sebuah *hardware* didalam jaringan, dan dapat di akses dari berbagai tempat melalui *ethernet*.

### 3.7.2 Konektifitas link SCADA

*PLC* yang telah diprogram pada sistem pengisian dan penutupan kaleng secara otomatis dapat dioperasikan dan ditempatkan dimana industri ataupun pabrik itu beroperasi. Sedangkan rencana pengontrolan dan monitoring *SCADA* akan dilakukan di *CCR (Central Control Room)*, bisa bertempat di lokasi yang sama, ataupun tempat yang berlokasi cukup jauh dari lokasi penempatan *PLC* yaitu diluar area industri atau pabrik. *SCADA* dihubungkan melalui suatu jaringan komputer *ethernet lan* dengan menggunakan **Hub**, dilakukan dengan menggunakan *Twisted Pair Cable* yang dihubungkan secara *pear to pear*. Ada dua buah *PC* yang digunakan untuk mengontrol dan memonitor jalannya sistem. Seperti halnya *PC* yang terhubung dalam suatu jaringan, agar dapat saling berkomunikasi haruslah diberikan **IP address** sebagai identitas, begitupun dengan

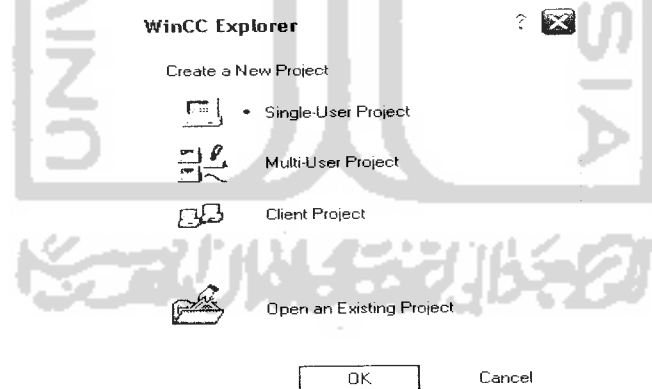
*PLC* yang terhubung dalam jaringan ini. *PLC* yang terhubung dianggap sebagai sebuah *PC*, maka *PLC* juga harus diberi *IP address*.

### 3.7.3 Pembuatan SCADA

Pembuatan *SCADA* (*Supervisory Control And Data Aquisition*) dapat dilakukan di HMI (*Human Machine Interface*), antara lain *InTouch Wonderware* dan *WinCC* (*Window Control Centre*) siemens. Pada kali ini pembuatan *SCADA* dilakukan di *Software WinCC*, karena semua *PLC* yang digunakan berasal dari *vendor* yang sama.

#### 3.7.3.1 Pembuatan project baru

Jika membuka *new project*, maka *WinCC* akan menampilkan beberapa pilihan yang dapat dipilih sesuai keinginan programmer.

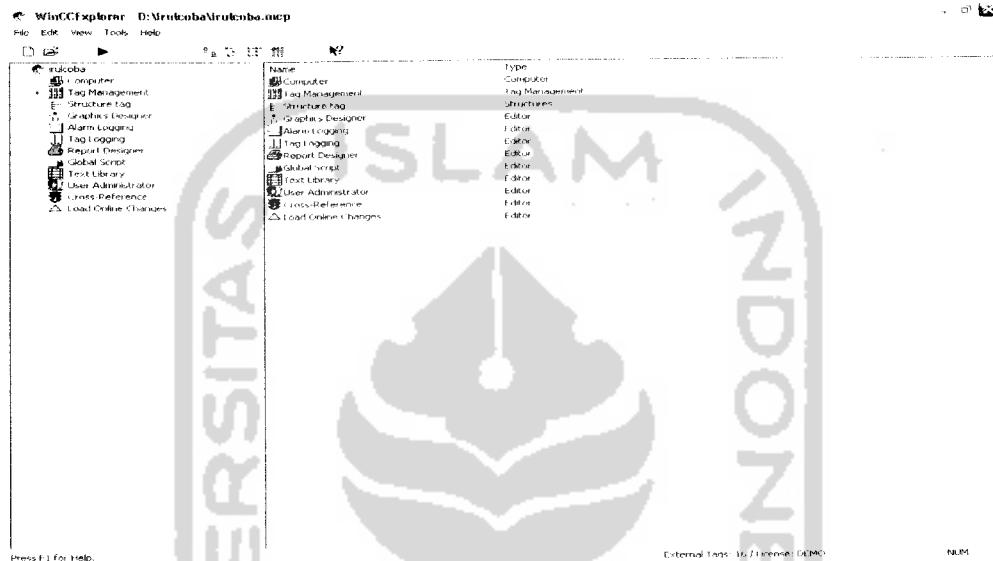


Gambar 3.17 Pembuatan project baru

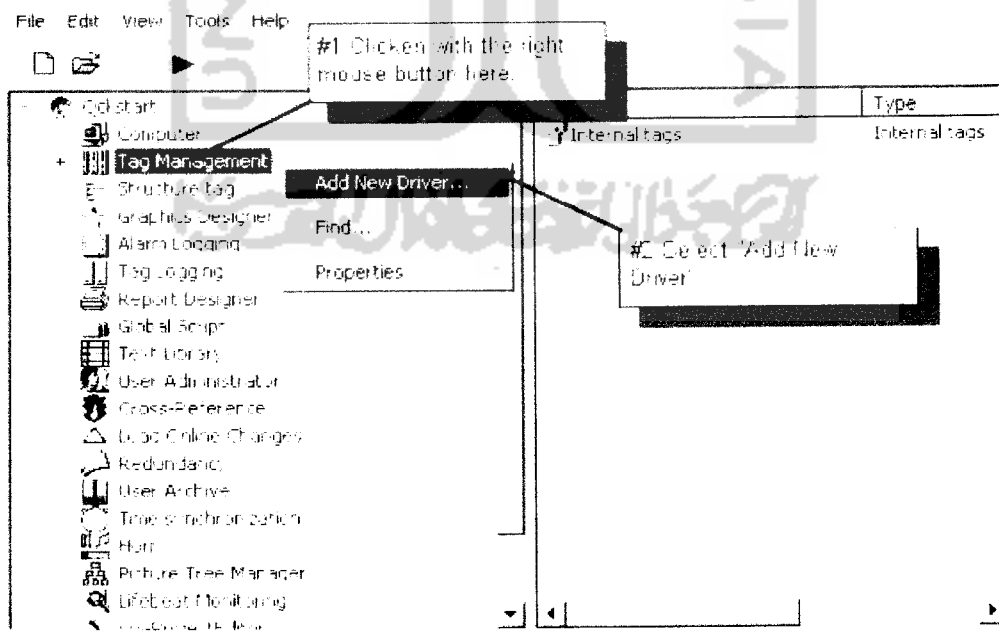
#### 3.7.3.2 Penambahan komunikasi driver

Untuk memudahkan akses antara *PLC* dengan *WinCC*, dibutuhkan driver komunikasi yang dapat ditambahkan dengan cara *mengklik* kanan pada *Tag*

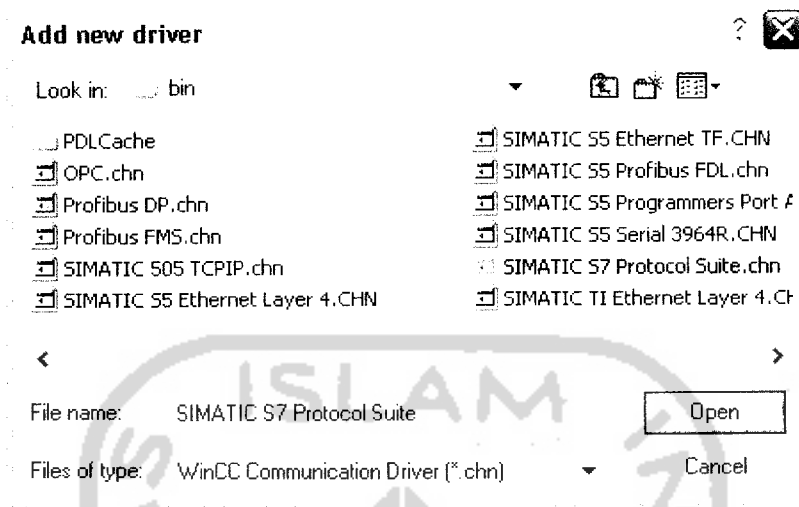
*Management*. Pada perancangan ini digunakan *S7 protocol suite.CHN*, sebagai sarana komunikasi antara *server PC* dengan *PLC*. Karena sesuai dengan jenis *PLC* yang digunakan. Untuk lebih memudahkan, perhatikan gambar dibawah ini.



Gambar 3.18 Tampilan WinCC

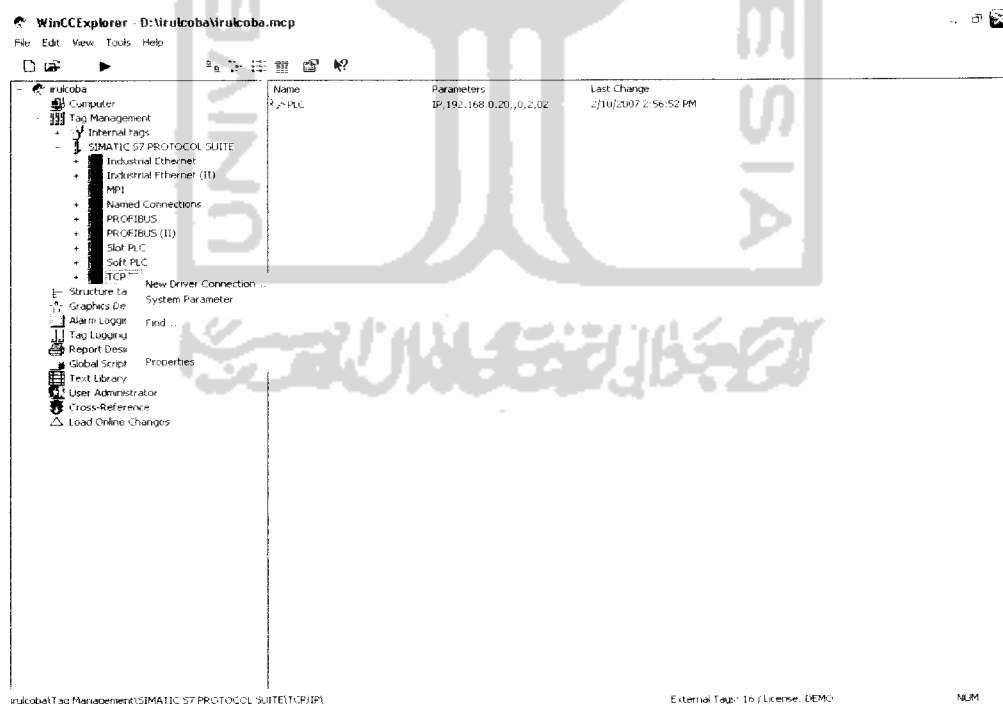


Gambar 3.19 Penambahan komunikasi driver



Gambar 3.20 Pilihan penambahan komunikasi driver

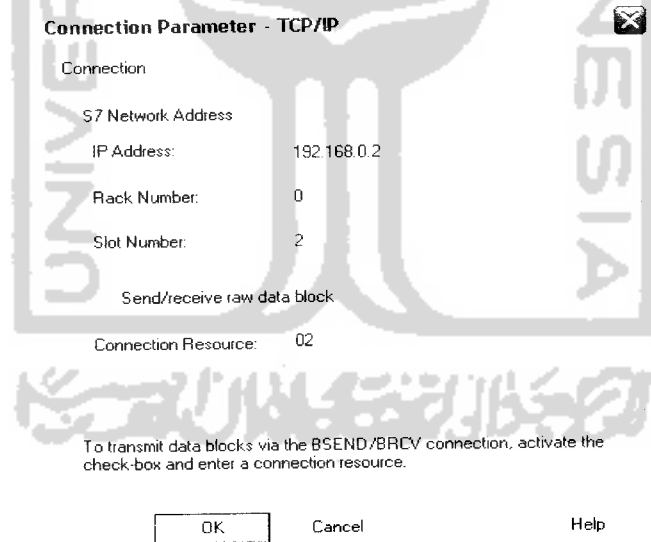
Pada *add new driver* dipilih *simatic protocol suite.chn*, karena sesuai dengan jenis PLC yang digunakan yaitu *S7-200 dan S7-300*.



Gambar 3.21 Penambahan komunikasi *driver* berdasarkan *interface*



Gambar 3.22 Connection properties

Gambar 3.23 Connection parameter- *TCP/IP*

Pada connection properties – TCP/IP digunakan alamat **IP standar** yaitu **192.168.0.2** dan **rack number 0** serta **slot number 2**. Untuk slot harus digunakan

slot nomor 2, karena PLC selalu di slot ini. Hal ini harus sesuai dengan *settingan* pada *CP 343-1 Lean* di *HW config Step 7 simatic manager*.

### 3.7.3.3 Pembuatan TAG

*Tag* yang digunakan di *WinCC* merupakan suatu nilai yang mewakili nilai yang sebenarnya. Untuk menyambungkan hubungan data yang berbeda antara *WinCC* dengan *PLC* dilakukan di eksternal *Tag*. Informasi-informasi mengenai *Tag* disimpan dalam *Tag Management* yaitu merupakan pusat informasi *Tag* bagi *WinCC*, yang berisi nilai aktual setiap *Tag* yang digunakan. Berikut ini hirarki *Tag Management* untuk melakukan proses *Tag*.

Tag Management Hierarchy with regard to Process Tags



Gambar 3.24. Hirarki Tag Management

Properties of tag group

General Tag



Name: Conveyor

Number of Tags: 16

Specify the name of the tag group

OK

Cancel

Help

Gambar 3.25 Penambahan tag group



Untuk membuat alamat pada *Tag*, di *Tag properties* ketik nama *Tag* yang akan ditampilkan. Setelah itu *klik select* untuk memasukkan alamat *Tag* tersebut, sesuai dengan I/O nya. Perhatikan Gambar 3.26, 3.27 dan 3.28.

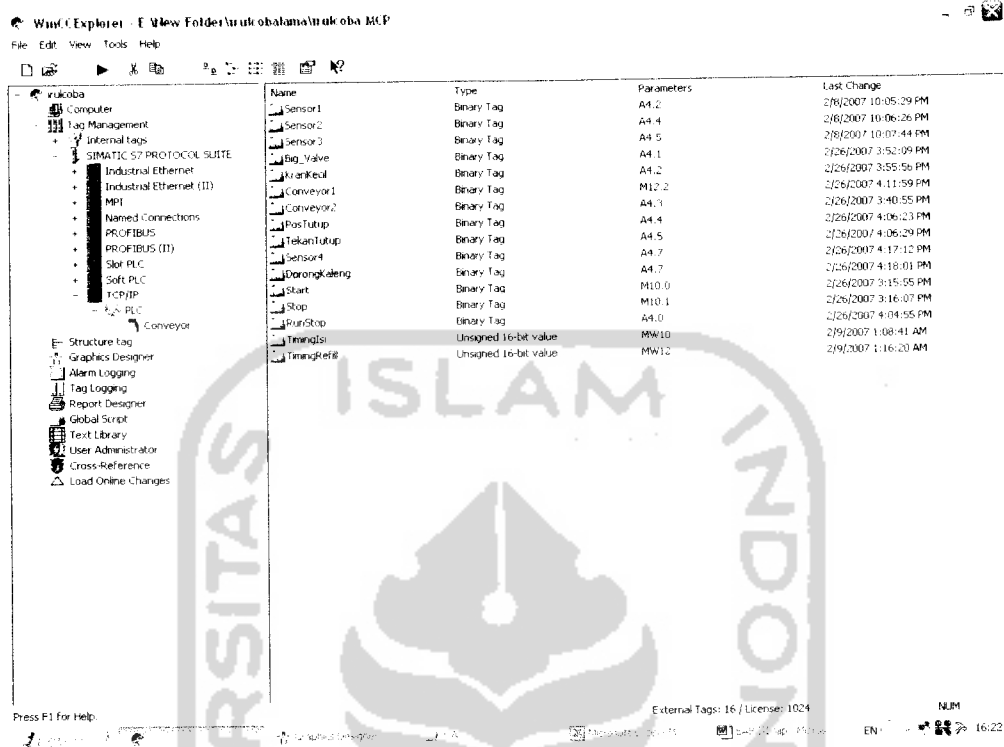


Gambar 3.26. Tampilan pembuatan Tag

Select the data area

OK Cancel Help

Gambar 3.27 Tampilan pembuatan alamat pada Tag

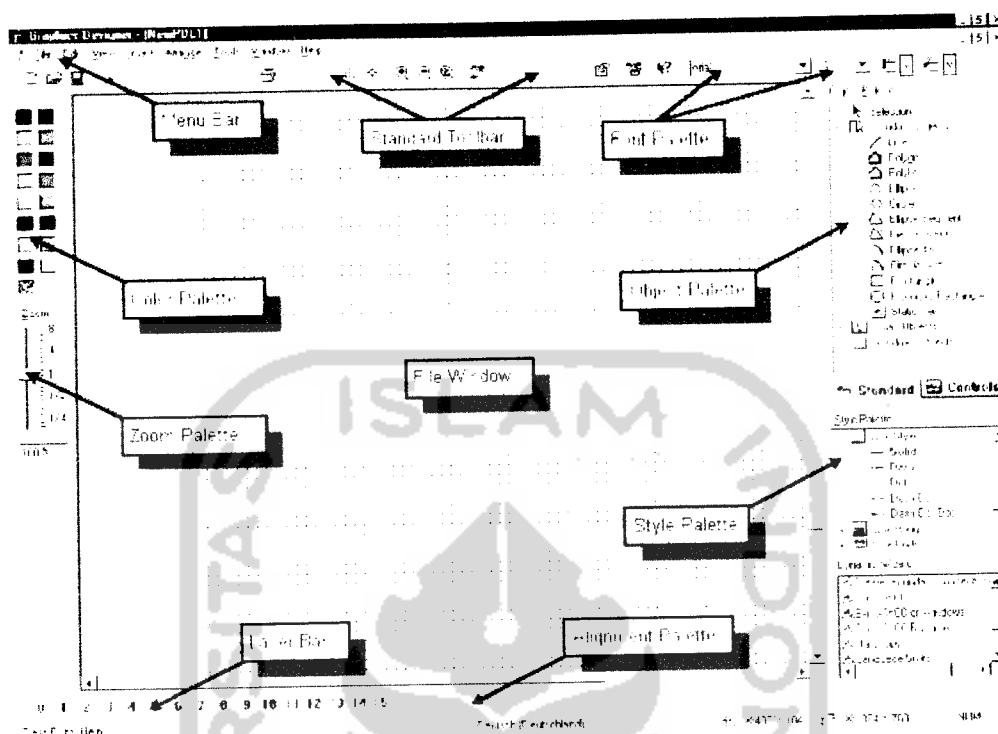


Gambar 3.28 Tampilan pembuatan Tag telah selesai

### 3.7.3.4 Pembuatan tampilan proses

Menggambar tampilan proses pada runtime dibuat dengan sistem grafis.

Pada bagian ini akan digambar tampilan proses yang akan ditampilkan.



Gambar 3.29 Pembuatan tampilan proses

Keterangan :

#### Menu Bar

Berisi seluruh *menu* yang dapat mengatur rancangan grafis.

#### Color Palette

Bertugas memilih warna dasar objek.

#### Zoom Pallette

Memilih faktor perbesaran tampilan (dalam persen) disaat *windows* aktif.

#### Layer Bar

Untuk memilih lapisan yang dapt ditampilkan hingga 32 lapisan.

#### Alignment palette

Untuk menggantikan posisi satu objek atau lebih.

### Style Palette

Mengganti tampilan pada objek yang dipilih.

### Object Palette

Berisi objek standar dan *windows* objek.

### Font Palette

Mengganti tipe huruf, ukuran dan warna seperti warna garis dari objek standar.

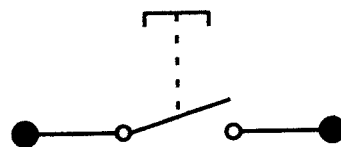
### Standard Toolbar

Berisi tombol untuk mempercepat melakukan perintah keadaan.

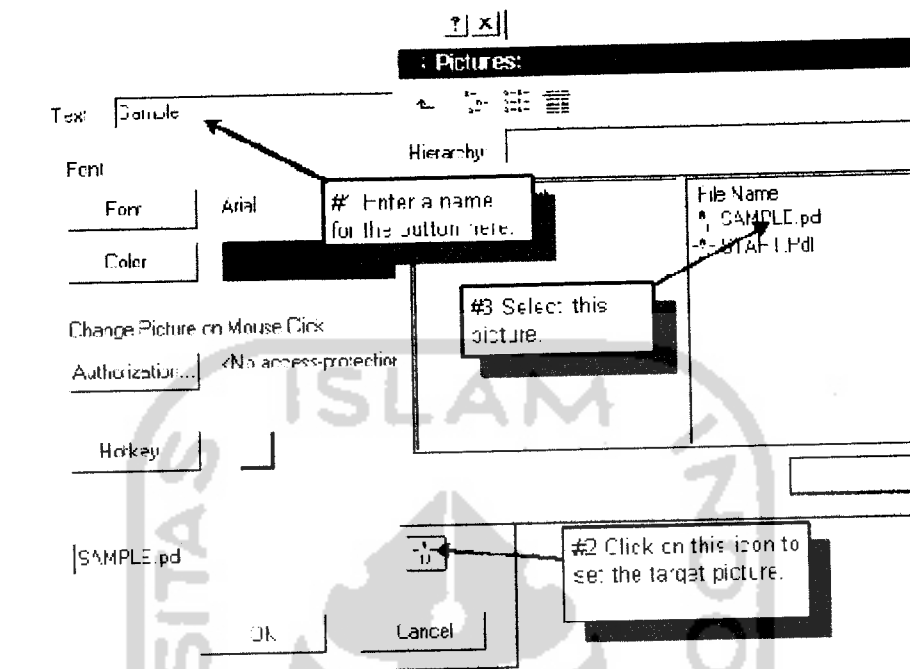
#### 3.7.3.5 Pembuatan tombol

Langkah pertama yaitu mengkonfigurasi tombol yang memungkinkan dapat menampilkan tampilan selain di *runtime*.

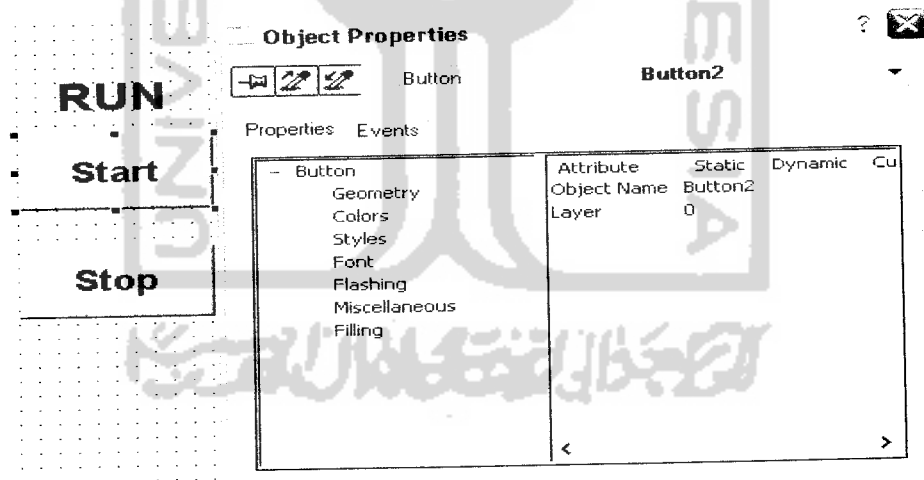
Untuk pengontrolan manual, tombol ataupun *push button* dijadikan alternatif pilihan. Digunakan dua buah *push button* untuk mengontrol *plant*-nya, yang pertama untuk mengaktifkan dan yang kedua untuk mematikan. Jenis *push button* yang digunakan adalah *push button normally open*, yang mana artinya apabila tombol ditekan maka saklar akan terhubung, dan akan terputus kembali ketika tombol dilepas. Berikut ini salah satu contoh gambar *push button NO*.



Gambar 3.30 Push button NO



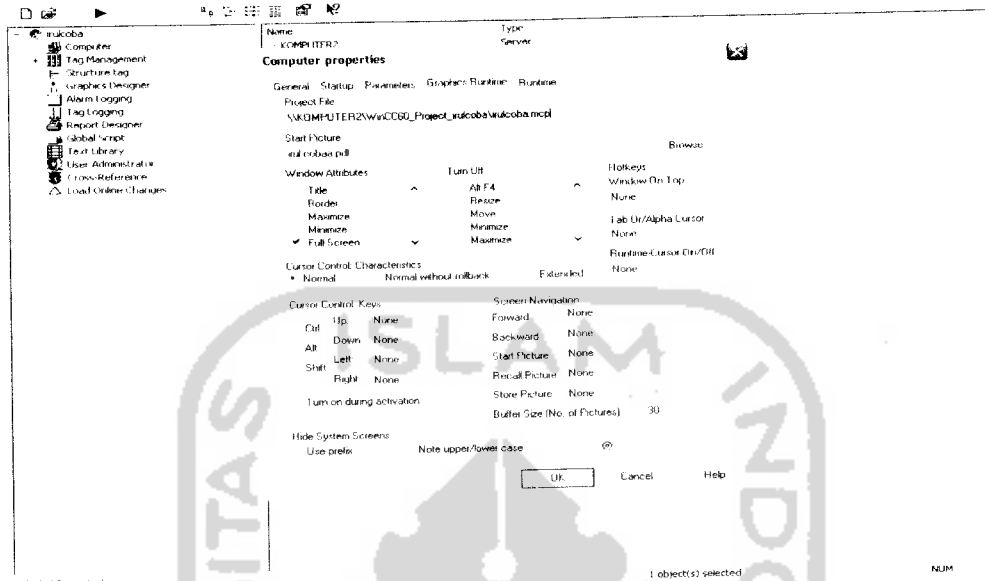
Gambar 3.31 Cara mengkonfigurasi tombol



Gambar 3.32 Tombol pada objek properties

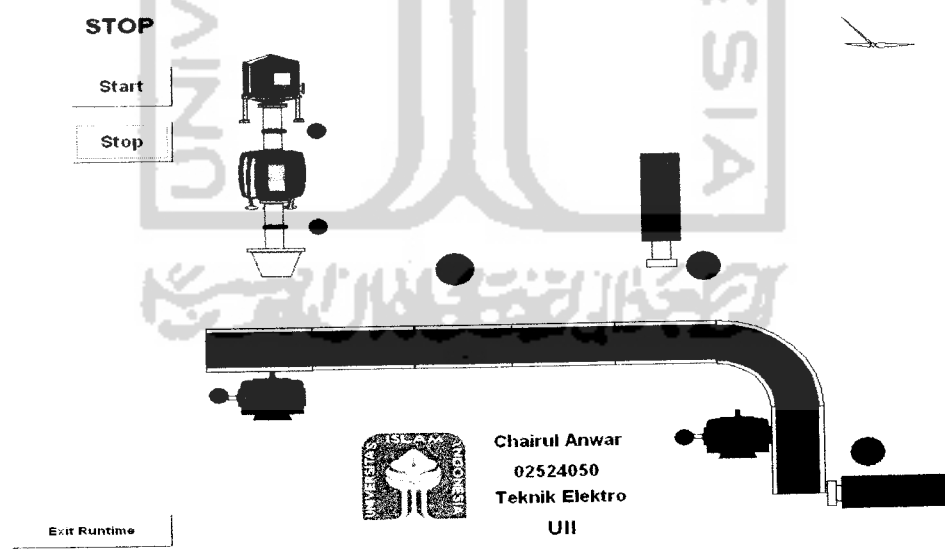
### 3.7.3.6 Mendefinisikan runtime

Runtime didefinisikan pada *komputer properties* di software WinCC untuk *project* yang dibuat, antara lain termasuk menampilkan *runtime*-nya.



Gambar 3.33 Definisi runtime pada *computer properties* di WinCC

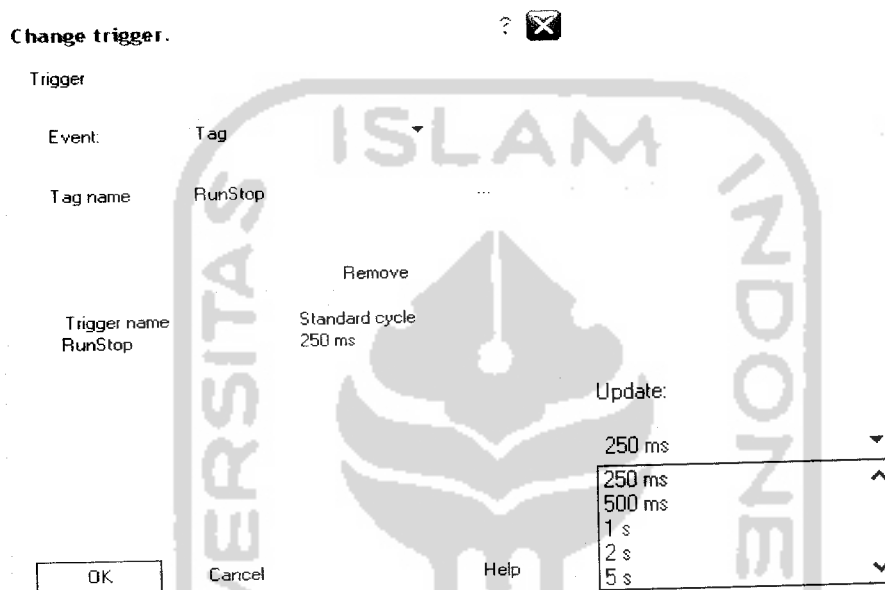
Aplikasi SCADA Sistem Pada Pengisian Dan Penutupan Kaleng Susu Secara Otomatis Berbasis PLC Siemens S7-200 Dan S7-300



Gambar 3.34 Tampilan *runtime*

### 3.7.3.7 Waktu tunda yang digunakan

Perputaran waktu yang digunakan antara *PC* dengan *plant* yaitu 250 ms. Sehingga antara *PC* dengan *plant* mempunyai penundaan waktu 250 ms. Untuk lebih jelasnya perhatikan Gambar 3.35 dan 3.36



Gambar 3.35 Penggunaan perputaran waktu

## BAB IV ANALISIS DAN PEMBAHASAN

Bab ini menjelaskan mengenai sistem *interface* antara dua *PLC*, yang mana lebih dikenal dengan sistem komunikasi *profibus*. Dan juga sistem pengisian, penutupan dan penempatan kaleng ke pinggir konveyor. Adapun materi pengujian sistem pengisian, dan penutupan kaleng meliputi ketepatan posisi kaleng pada tiap sensor dan kesamaan berat yang diisikan ke dalam kaleng.

### 4.1 Sistem komunikasi Profibus

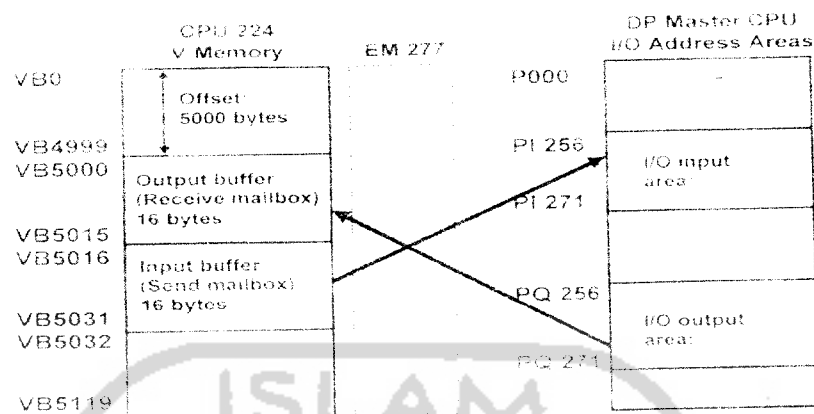
Simatic S7-200 dapat dihubungkan ke profibus-DP menggunakan EM 277 (*Expansion Module*). *Profibus-DP* adalah "*PROcess Field BUS*" dan DP adalah *Distributed Peripheral* yaitu pembagian sekitarnya yang menggunakan pengontrol.

Profibus-DP sangat erat hubungannya dengan pertukaran data. Untuk memudahkan sistem pertukaran data, perhatikan Gambar 4.1 *V memory*. Berikut ini ilustrasi pertukaran data pada *V memory* :

1. *Master* menulis data ke *slave* dan merespon permintaan *input* data.
2. *Master* mengirim data dari I/O output ke *ouput Buffer slave* (menerima kotak pos).
3. *Master* juga membaca data dari *input buffer slave* (mengirim kotak pos).

Sehingga antara *master* dan *slave* dapat saling membaca *input* maupun *outputnya*.





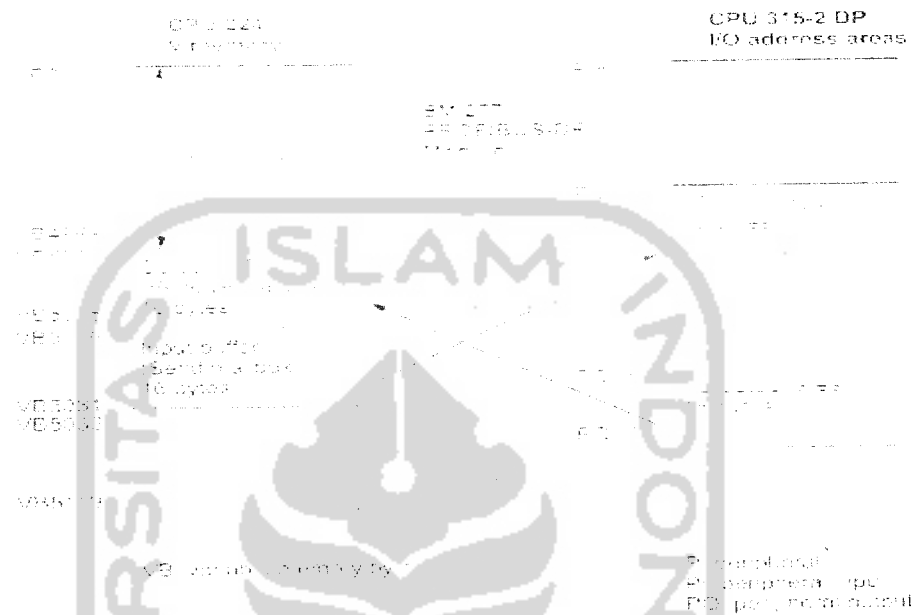
Gambar 4.1 V memory

*Profibus-DP* selalu menggambarkan pemindahan data dari *master* ke *slave* ataupun sebaliknya. Data yang dikirim dari *master* ke *slave* selalu menunjukkan *input* data. Didalam *slave* nilai data yang datang dari *master* masih menunjukkan *output* data, meskipun input mereka ke *slave*. Pada cara yang sama, nilai data dikembalikan ke *master* yang masih menunjukkan *input*, walaupun *output* mereka dari *slave*.

*Master* menerangkan ke *slave* untuk memulai alamat ke *output* buffer. Jika *offset* adalah 0, *slave* akan menempatkan penyangga input (*input buffer*) pada VB0. Jika *offset* adalah 5000, *slave* akan menempatkan penyangga output (*output buffer*) pada VB5000.

*Master* juga menerangkan ukuran *slave* di *output buffer* menggunakan konfigurasi *master* dan menulis beberapa nomor *bytes* dari data ke *slave*. *Master* mengirim informasi ini ke *slave* sebagai bagian dari konfigurasi *slave*. *Slave* menggunakan informasi ini untuk mengatur ukuran *output buffer*. Jika *master* menerangkan *slave* ada 16 *output bytes*, maka *slave* akan mengatur *output buffer* jadi 16 *bytes*. Namun jika *output buffer* dimulai pada VB5000, maka output data

dari *master* juga akan menulis dilokasi VB5000 sampai dengan VB5015. Untuk lebih jelasnya perhatikan gambar 4.2 sistem pertukaran data pada V memory.



Gambar 4.2 Sistem pertukaran data pada V memory

#### 4.2 Standar Komunikasi DP (Distibuted Peripheral)

*Modul EM 277 Profibus-DP* memiliki protokol standar yang mengartikan *slave* terbagi menjadi dua, yaitu :

a. EN 50 170 (PROFIBUS)

Mengambarkan akses bus dan mengirimkan protokol serta *properties* yang lebih khusus dari data pengiriman *medium*.

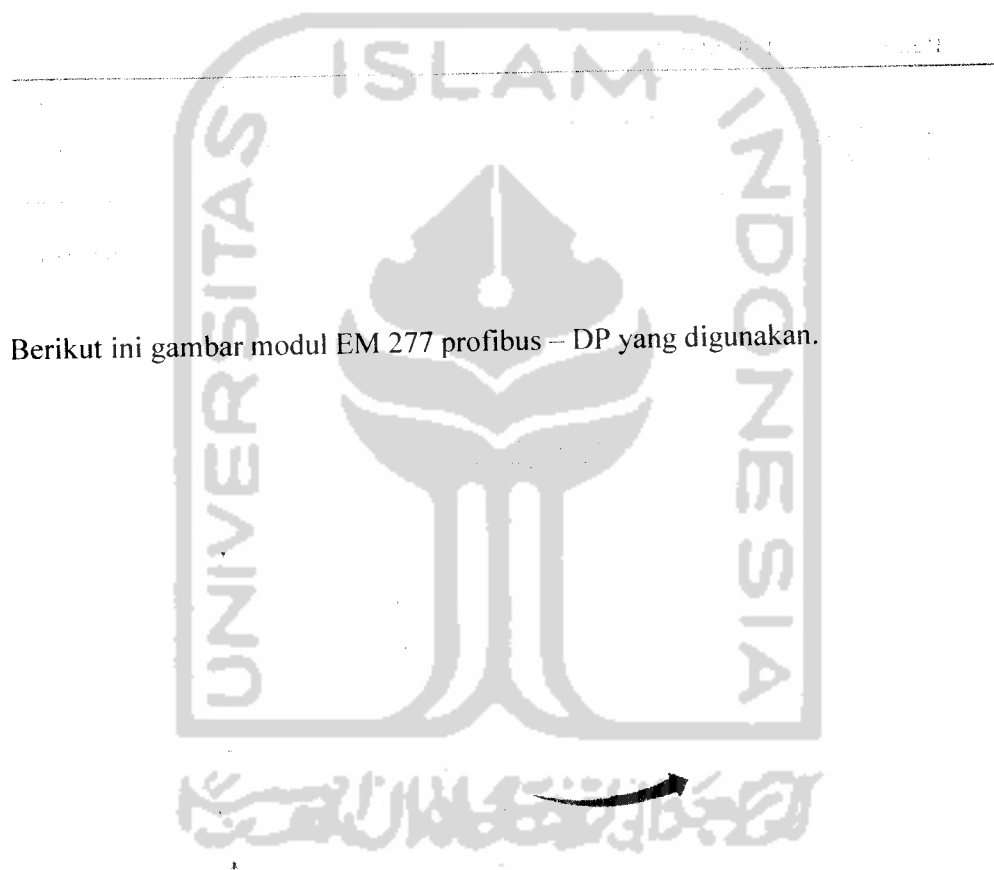
b. EN 50 170 (DP Standard)

Menggambarkan pertukaran perputaran data kecepatan tinggi antara *master* DP dan *slave* DP.

### 4.3 Indikator status EM 277 Profibus-DP

Modul EM 277 profibus-DP memiliki empat indikator di depan panel yang menunjukkan sistem operasionalnya. Untuk lebih mudah dalam memahaminya, maka perhatikan table 4.1 berikut ini.

Tabel 4.1 Status LED modul EM 277 profibus - DP



Berikut ini gambar modul EM 277 profibus – DP yang digunakan.

Gambar 4.3 Contoh gambar EM 277 Profibus – DP



Gambar 4.4 Sistem jaringan profibus

#### 4.4 Pengujian Sistem Pneumatik

Katup dan *cylinder* memiliki ambang batas tekanan yang mampu untuk ditampung sebesar 10 bar.

Tabel 4.2 Pengujian tekanan sistem pneumatik

Tekanan (bar)	Gaya dorong
1	Tidak ada
2	Terlalu kecil
3	Baik
4	Terlalu Besar
5	Terlalu Besar
6	Terlalu Besar
7	Terlalu Besar
8	Terlalu Besar
9	Terlalu Besar
10	Terlalu Besar

Bar merupakan satuan tekanan yang digunakan pada sistem pneumatik, dimana  $1 \text{ bar} = 1.10^5 \text{ N/m}^2$

Dari hasil pengamatan didapatkan bahwa :

1. Pada tekanan 1 bar sampai 2 bar tidak terjadi gaya dorong dikarenakan tekanan terlalu kecil.
2. Pada tekanan 3 bar gaya dorong dari piston bekerja dengan baik. Hal ini dikarenakan karena hentakan yang dihasilkan oleh tekanan 3 bar tidak terlalu besar dan tidak terlalu kecil.
3. Pada tekanan  $> 3$  bar gaya dorong dari piston terlalu besar.

#### 4.5 Pengujian ketepatan posisi kaleng pada masing-masing sensor

Pada percobaan ini dilakukan pengujian ketepatan posisi berhenti terhadap 5 kaleng tanpa melakukan proses pada masing-masing sensor. Hal ini dilakukan untuk membandingkan ada tidaknya pengaruh dari hentakan piston akibat tekanan udara dari kompresor.

Kaleng yang berhenti dengan tepat dinyatakan dengan angka 1 dan kaleng yang berhenti tidak tepat pada tempatnya dinyatakan dengan angka 0.

Tabel 4.3 Pengujian sensor dengan jarak antara setiap kaleng renggang( 5 cm )

Kaleng	Konveyor 1			Konveyor 2
	Sensor 1	Sensor 2	Sensor 3	Sensor 1
Kaleng 1	1	1	1	1
Kaleng 2	1	1	1	1
Kaleng 3	1	1	1	1
Kaleng 4	1	1	1	1
Kaleng 5	1	1	1	1

Tabel 4.4 Pengujian sensor dengan jarak antara setiap kaleng rapat ( 2 cm )

Kaleng	Konveyor 1			Konveyor 2
	Sensor 1	Sensor 2	Sensor 3	Sensor 1
Kaleng 1	1	1	1	1
Kaleng 2	1	1	1	1
Kaleng 3	1	1	1	1
Kaleng 4	1	1	1	1
Kaleng 5	1	1	1	0

Tabel 4.5 Pengujian sensor dengan jarak antara setiap kaleng acak

Kaleng	Konveyor 1			Konveyor 2
	Sensor 1	Sensor 2	Sensor 3	Sensor 1
Kaleng 1	1	1	1	1
Kaleng 2	1	1	1	1
Kaleng 3	1	1	1	0
Kaleng 4	1	1	1	1
Kaleng 5	1	1	1	0

Pada pengujian ketepatan pemberhentian kaleng pada masing-masing sensor pada konveyor 1 berhasil dengan baik sedangkan pada konveyor 2 tidak semua kaleng berhenti dengan tepat hal ini disebabkan pada motor konveyor 1 dipasang rem sedangkan pada motor konveyor 2 tidak dipasang rem.

#### 4.6 Pengujian berat bubuk

Pegujian berat bubuk yang diisikan ke kaleng bertujuan untuk melihat kinerja sensor berat yang digunakan pada sistem pengisian.

Tabel 4.6 Pengujian Berat Bubuk

Kaleng	Berat (gram )
1	0,65
2	0,63
3	0,65
4	0,64
5	0,65

Adapun nilai rata-rata dan standar deviasinya yaitu:

a. Nilai rata-rata pengujian

$$\begin{aligned}
 &= \sum_{n=1}^N \text{Berat}/N \\
 &= \frac{0,65 + 0,63 + 0,65 + 0,64 + 0,65}{5} \\
 &= 0,644 \text{ gram}
 \end{aligned}$$

b. Deviasi standar

$$\sigma^2 = \frac{(0,65 - 0,64)^2 + (0,63 - 0,644)^2 + (0,65 - 0,644)^2 + (0,64 - 0,644)^2 + (0,65 - 0,644)^2}{5}$$

$$\sigma^2 = \frac{3,2 \times 10^{-4}}{5}$$

$$\sigma^2 = 6,4 \times 10^{-5}$$

$$\sigma = \sqrt{6,4 \times 10^{-5}}$$

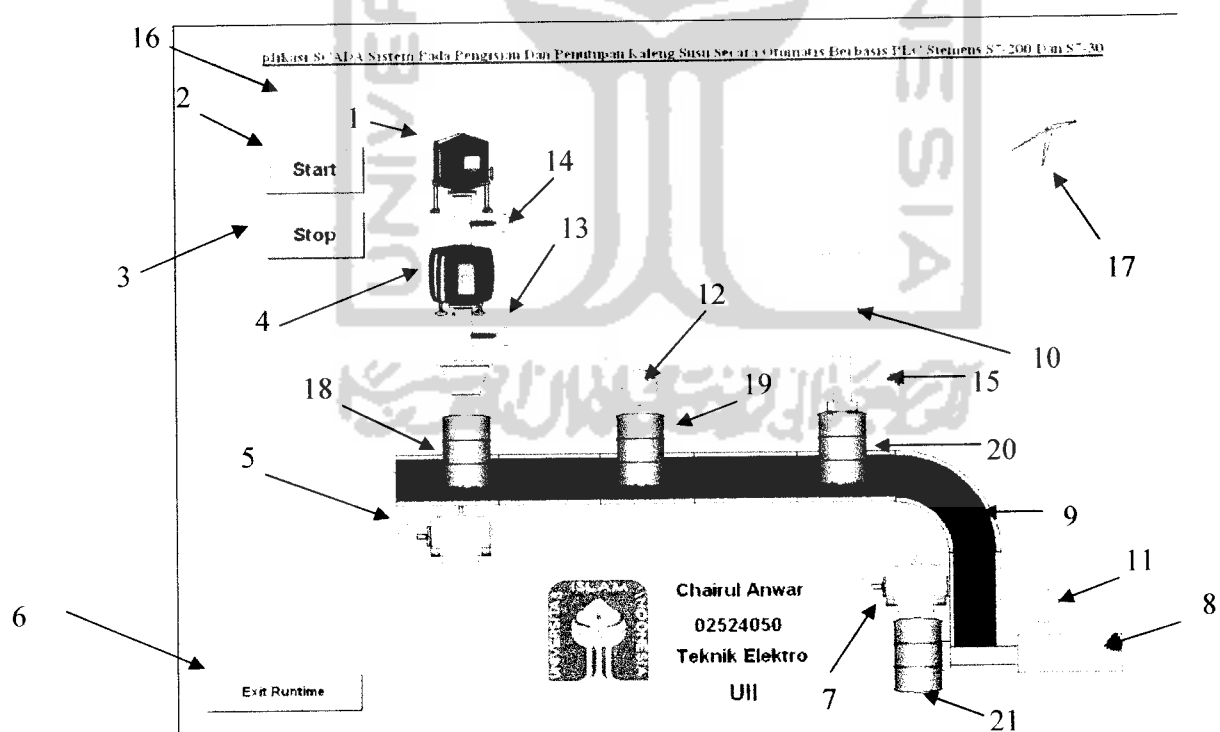
$$\sigma = 0,008 \text{ gram}$$

Jadi nilai rata-rata dan standar deviasinya yaitu: 0,644 gram dan 0,008 gram.

Dari hasil pengujian didapatkan bahwa penggunaan sensor berat dalam sistem pengisian belum akurat.

#### 4.7 Pemrograman SCADA

Tampilan display SCADA pada layar monitor masing – masing PC yang diberikan akses untuk mengontrol PLC adalah tidak harus selalu sama. Tampilan display dapat diubah – ubah sesuai dengan yang diinginkan, asalkan alamat dan *tag* dari masing – masing komponen harus sama. Namun pada intinya display yang ditampilkan haruslah mudah untuk dimengerti, mudah untuk dioperasikan dan bisa menggambarkan bagaimana kondisi sebenarnya yang terdapat di lapangan. Pada perancangan tampilan untuk *SCADA* kali ini menggunakan dua buah saklar *push button*, dan tampilan – tampilan keadaan yang digambarkan dengan pewarnaan. Untuk lebih jelasnya lihat gambar 4.5 tampilan *SCADA*.



Gambar 4.5 Tampilan mimic diagram SCADA untuk pengisian dan penutupan kaleng susu



Pengoperasian dari animasi diatas cukup mudah, dengan ini operator dapat menghidupkan dan mematikan *plant* hanya dengan menekan tombol *start* dan *stop*.

**Keterangan :**

1. Tangki Besar  
Menggambarkan persediaan susu yang masih tersedia.
2. Tombol Start  
Tombol yang berfungsi untuk menjalankan atau mengoperasikan plant.
3. Tombol Stop  
Tombol yang berfungsi untuk mematikan ataupun menonaktifkan plant.
4. Tangki Kecil  
Menggambarkan berat jumlah susu yang akan dituangkan kedalam kaleng.
5. Motor konveyor Satu  
Menggambarkan motor konveyor satu yang bekerja atau online.
6. Exit Runtime  
Berfungsi sebagai tombol keluar runtime ataupun online.
7. Motor Konveyor Dua  
Menggambarkan motor konveyor dua yang bekerja atau online.
8. Pneumatik Memindahkan Kaleng  
Menggambarkan sistem pemindahan kaleng dari konveyor 2.

9. Konveyor  
Menggambarkan alat pembawa barang atau kaleng.
10. Pneumatik Penekanan Tutup  
Menggambarkan sistem penekanan tutup kaleng.
11. Indikator Sensor 4  
Menandakan ada tidaknya kaleng yang terdeteksi untuk dipindahkan dari konveyor.
12. Indikator Penempatan Tutup.  
Mengilustrasikan penempatan tutup kaleng.
13. Indikator Pembuka Kran Kecil  
Mengindikasikan pembuka pembuka kran atau valve kecil.
14. Indikator Pembuka kran besar  
Mengindikasikan pembuka kran atau valve besar.
15. Indikator Sensor 3  
Menandakan ada tidaknya kaleng yang terdeteksi pada waktu penekanan tutup kaleng.
16. Run Stop  
Menandakan kondisi plant sedang aktif atau on dan juga menandakan kondisi tidak aktif atau off.
17. Waktu (jam)  
Menandakan waktu saat plant akan dijalankan.
18. Posisi Kaleng 1  
Menandakan kondisi kaleng saat pengisian bubuk atau susu.

19. Posisi Kaleng 2

Menandakan kondisi kaleng saat penempatan tutup.

20. Posisi Kaleng 3

Menandakan kondisi kaleng saat penekanan tutup.

21. Posisi Kaleng 4

Menandakan kondisi kaleng telah dipindahkan dari konveyor 2.

#### 4.8 Analisa kerja sistem secara keseluruhan

Keseluruhan sistem berjalan cukup baik. Secara umum kinerja dari keseluruhan sistem juga terbilang sangat mudah, hal ini ditunjang animasi yang terdapat pada pemrograman *SCADA*, sehingga sangat mendekati dengan kondisi yang sebenarnya ataupun pada *plant*. Mengaktifkan saklar pada *plant* sama juga artinya dengan mengaktifkan sistem melalui saklar pada pemrograman *SCADA*. Setiap aktifitas sistem terbaca dengan baik pada program *SCADA* yang telah dirancang,

Pengujian sistem pengisian, dan penutupan kaleng meliputi ketepatan posisi kaleng pada tiap sensor dan kesamaan berat yang diisikan kedalam kaleng. Ini dilakukan untuk melihat keberhasilan sistem dalam menjalankan prosesnya. Pengujian ini menggunakan 3 metode, yaitu:

1. Penempatan kaleng dengan jarak antara setiap kaleng renggang ( 5 cm )
2. Penempatan kaleng dengan jarak antara setiap kaleng rapat ( 2 cm )
3. Penempatan kaleng secara acak

Proses yang berhasil dinyatakan dengan angka 1 dan proses yang gagal dinyatakan dengan angka 0.

Dari hasil percobaan dapat diamati kegagalan proses banyak terjadi pada proses penutupan kaleng. Kegagalan proses disebabkan kurang tepatnya peletakan tutup pada mulut kaleng sehingga pada saat tutup ditekan tutup tidak berada pada posisi yang tepat. Agar proses penutupan ini mempunyai tingkat keberhasilan yang lebih tinggi maka diperlukan mekanisme penutupan yang lebih akurat.



## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dari hasil pengujian yang dilakukan maka dapat ditarik kesimpulan bahwa:

1. Jarak antar kaleng harus lebih dari 1 cm untuk meminimalisasi kegagalan pada saat penekanan tutup kaleng, karena diameter plat penutup kaleng lebih besar 0,5 cm dari diameter tutup kalengnya.
2. Transmisi data antara *master* dan *slave* terbagi kedalam 3 fase yaitu: *parameter assignment* (ukuran penempatan), *konfigurasi*, dan *data exchange* (pertukaran data).
3. *SCADA* memberikan wujud visualisasi dari sistem yang telah dibangun melalui serangkaian bentuk *display animasi mimic diagram* yang berkaitan langsung dengan alamat *I/O*, dan dirancang sehingga menyerupai kondisi pada *plant*.
4. *WinCC* merupakan perangkat lunak yang berfungsi menjembatani hubungan antara *PLC* dengan *PC* dengan mengaplikasikan program untuk menampilkan *SCADA*.

## 5.2 Saran

1. Pada penelitian berikutnya dicoba dengan PLC tipe dan jenis yang lain, agar terlihat perbedaan dan tingkat kerumitan pada perancangan SCADA berikutnya.
2. Untuk mendapatkan hasil kerja yang memuaskan, hendaknya harus memiliki banyak referensi dan pengalaman yang cukup dalam pembuatan SCADA.
3. Penggunaan software WinCC untuk pembuatan program SCADA relatif lebih mudah dan mampu menunjang untuk semua jenis PLC.
4. Sensor berat yang dipakai dicoba diganti dengan sensor volume, agar berat dari setiap kaleng menjadi sama.
5. Untuk mendapatkan hasil yang maksimal dalam proses pengisian maka diperlukan pengukur berat takaran yang mempunyai tingkat ketelitian lebih tinggi dan untuk menghindari tumpahnya bubuk pada saat pengisian diperlukan saluran pengisian yang lebih sempurna.

## DAFTAR PUSTAKA

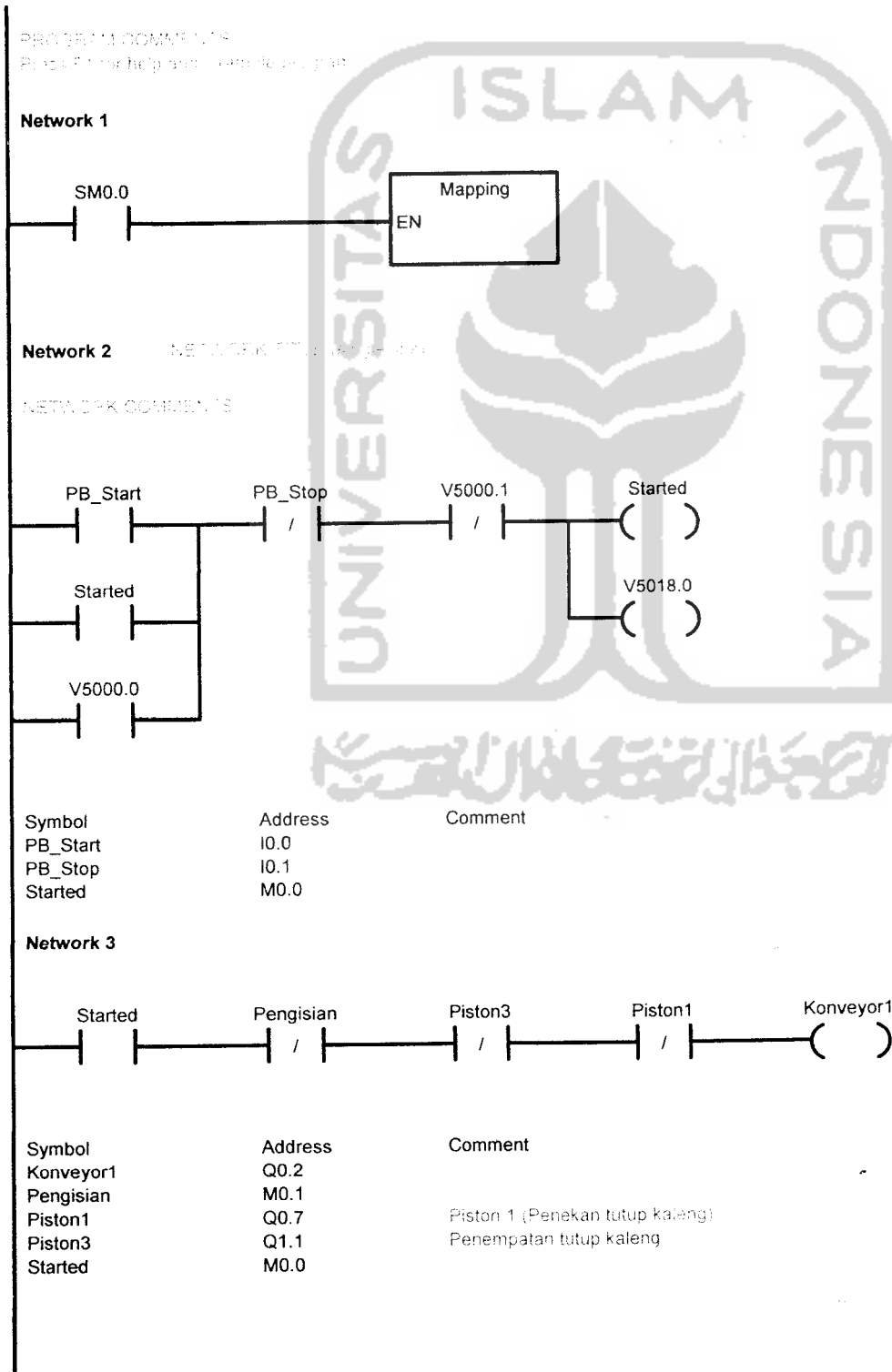
1. Ismara, Juke, 2005. "*InTouch – Human Machine Interface*". Biro Rancang Bangun, Departemen Rancang Bangun dan Rekayasa. PT Semen Padang. Sumatra Barat.
2. Ismara, Juke, 2005. "*Implementasi SCADA*". Biro Rancang Bangun, Departemen Rancang Bangun dan Rekayasa. PT Semen Padang. Sumatra Barat Indonesia.
3. AG, Siemens, 2003. "*Simatic HMP*". Siemens Germany.
4. AG, Siemens, 2005. "*S7-200 Specifications*". Siemens Germany.
5. AG, Siemens, 2005. "*Profibus DP Networkon Simatic S7-200*". Siemens Germany.
6. AG, Siemens, 2005. "*Profibus DP Communication*". Siemens Germany.
7. AG, Siemens, 2005. "*Manual Book S7-200*". Siemens Germany.
8. Ansori, Ato., 2006. "*Basic Principle of SCADA*". Makalah, tidak diterbitkan. Email : automationconsultant@gmail.com
9. El Arsyad, Fauzan Upaya., 2006. "*Perancangan Sistem Kontrol Berbasis SCADA Dengan Menggunakan Ethernet Pada PLC Untuk Pengaturan Lampu Gedung*". Skripsi, tidak diterbitkan. Jogjakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.
10. Setyawan, Heri., 2006. "*Perancangan Mekanisme Pengisian dan Penutupan Kaleng Susu Secara Otomatis Berbasis PLC*". Skripsi, tidak diterbitkan. Jogjakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.



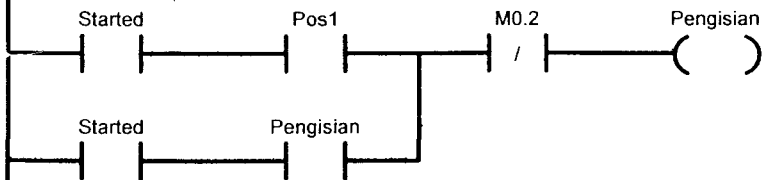


Block: Program\_Block  
 Author:  
 Created: 05/11/2006 12:30:28 pm  
 Last Modified: 02/16/2007 10:54:39 am

Symbol	Var Type	Data Type	Comment
	TEMP		
	TEMP		
	TEMP		
	TEMP		

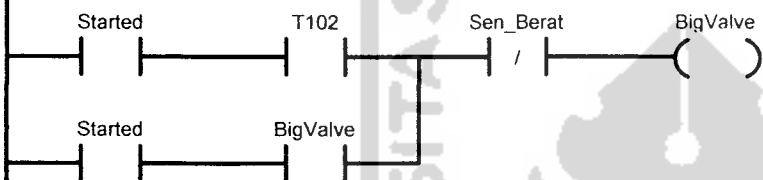


**Network 4**



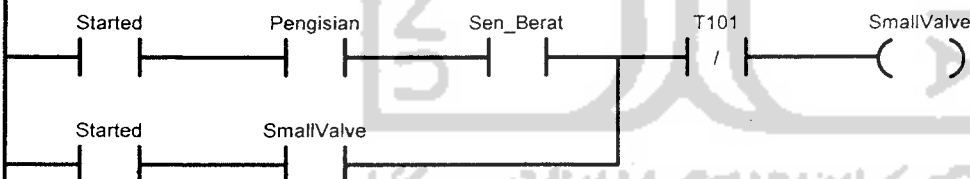
Symbol	Address	Comment
Pengisian	M0.1	
Pos1	I0.2	
Started	M0.0	

**Network 5**

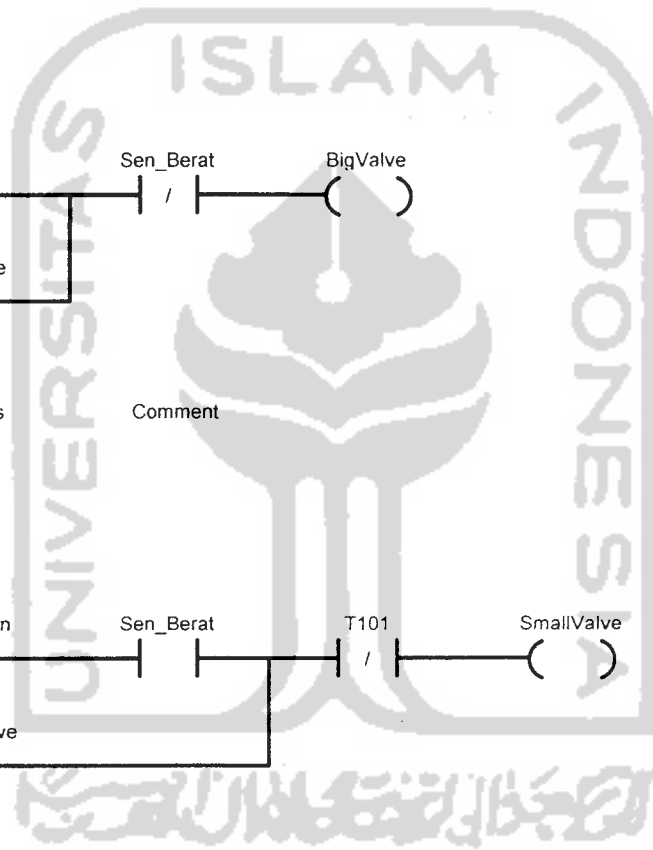


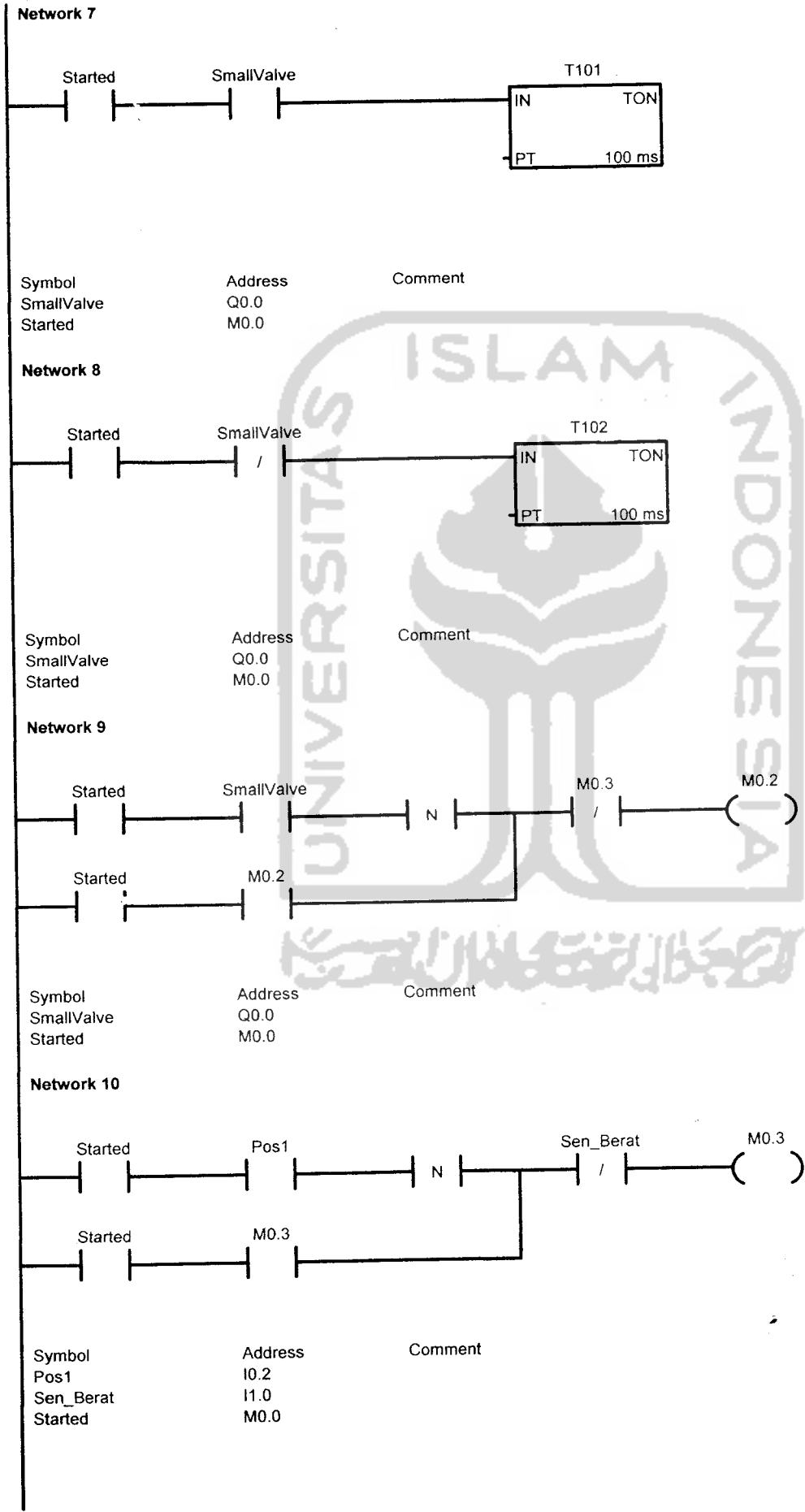
Symbol	Address	Comment
BigValve	Q0.1	
Sen_Berat	I1.0	
Started	M0.0	

**Network 6**

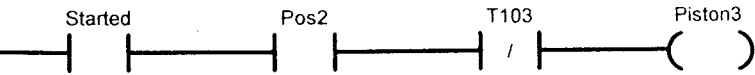


Symbol	Address	Comment
Pengisian	M0.1	
Sen_Berat	I1.0	
SmallValve	Q0.0	
Started	M0.0	





**Network 11**



Symbol	Address	Comment
Piston3	Q1.1	
Pos2	I0.4	
Started	M0.0	

**Network 12**



Symbol	Address	Comment
Pos2	I0.4	
Started	M0.0	

**Network 13**



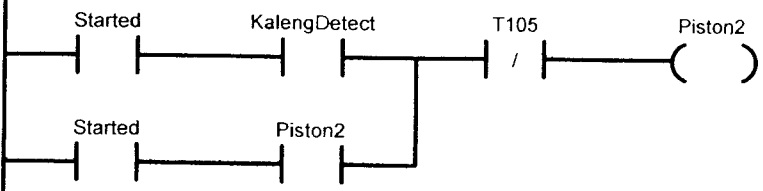
Symbol	Address	Comment
Piston1	Q0.7	
Pos3	I0.5	
Started	M0.0	

**Network 14**



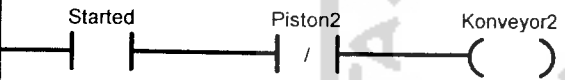
Symbol	Address	Comment
Pos3	I0.5	
Started	M0.0	

**Network 15**



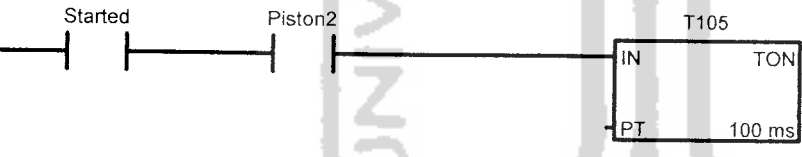
Symbol	Address	Comment
KalengDetect	I0.7	
Piston2	Q1.0	
Started	M0.0	

**Network 16**

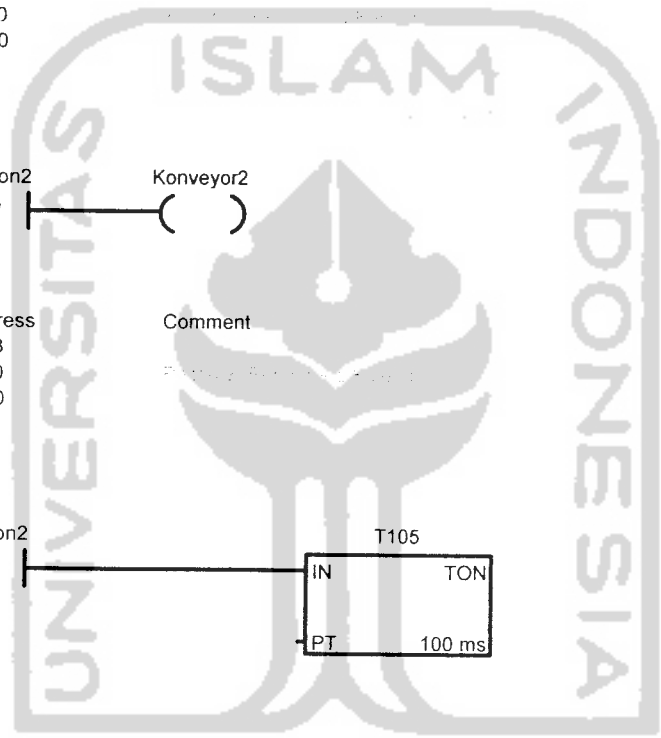


Symbol	Address	Comment
Konveyor2	Q0.3	
Piston2	Q1.0	
Started	M0.0	

**Network 17**



Symbol	Address	Comment
Piston2	Q1.0	
Started	M0.0	



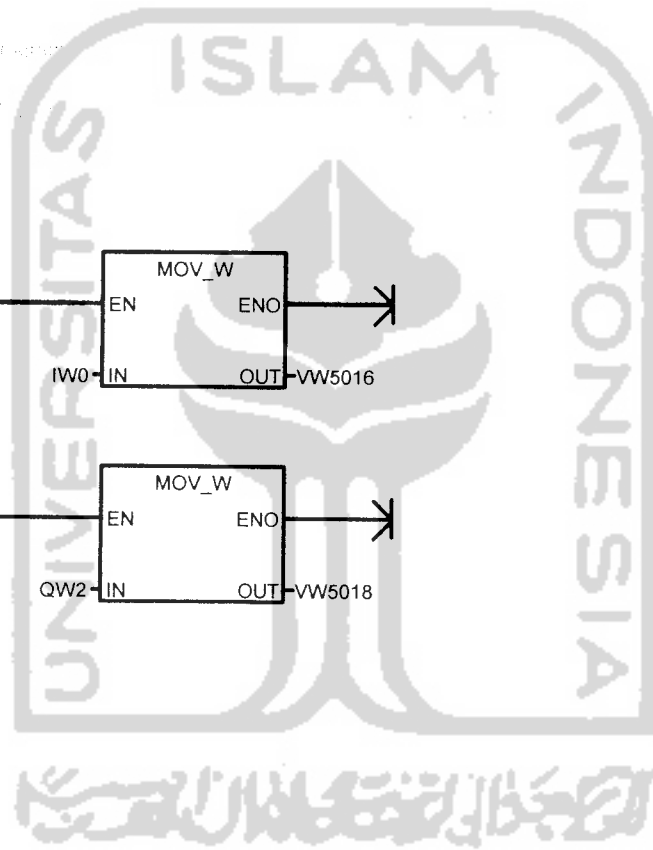
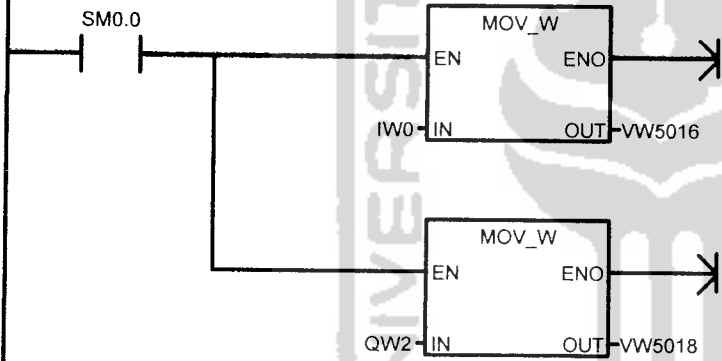
Block: Mapping  
 Author:  
 Created: 08/23/2006 10:22:22 am  
 Last Modified: 02/13/2007 09:19:01 am

Symbol	Var Type	Data Type	Comment
EN	IN	BOOL	
	IN		
	IN_OUT		
	OUT		
	TEMP		

Sub-Program COMMENTS  
 Please fill for help and example of work

Network 1

NETWORK COMMENTS



Block: INT\_0  
Author:  
Created: 08/23/2006 10:22:22 am  
Last Modified: 08/23/2006 10:54:05 am

Symbol	Var Type	Data Type	Comment
	TEMP		
	TEMP		
	TEMP		
	TEMP		

Network 1



Block: ETH0\_CTRL  
 Author: Ethernet Wizard  
 Created: 02/01/2006 01:10:51 pm  
 Last Modified: 02/01/2006 01:10:51 pm

	Symbol	Var Type	Data Type	Comment
	EN	IN IN IN_OUT	BOOL	
L0.0	CP_Ready	OUT	BOOL	CP 243-1 Module is ready
LW1	Ch_Ready	OUT	WORD	Channel ready bits
LW3	Error	OUT OUT TEMP	WORD	Error word



This POU was generated by the Ethernet Wizard for use with a CP 243-1 module. It is intended for use with the CP 243-1 module. This instruction is used to enable and disable the CP 243-1 module. This instruction is only used once in your program. The command type for this module was selected as 'OUT'.

No connections have been configured for this module.





## TIC 300(1)

## Rack 0

Port designation: UR  
 Order no.: 6ES7 330-1???0-0AA0  
 Designation: UR

## . 0, Slot 2

Port designation: CPU 313C-2 DP  
 Order no.: 6ES7 313-6CE01-0AB0  
 Designation: CPU 313C-2 DP  
 Width: 1  
 I address: 2  
 Highest MPI address: 31  
 baud rate: 187.5 Kbps  
 Comment:  
 ---

## Startup

Startup when expected/actual  
 config. differ: Yes  
 Hardware test at warm restart: No  
 Startup after Power On: Warm Restart  
 Monitoring time for  
 "Finished" message by means of  
 modules: 65000 ms  
 Transfer of the parameters  
 to modules: 10000 ms

## Scan cycle

Scan cycle monitoring time: 150 ms  
 Scan cycle load from  
 Communication: 20 %  
 OB85 call up if I/O access error: No OB85 call up

## Lock memory:

No

## Sensitivity

Number of memory bytes starting  
 at MB0: 16  
 Number of S7 timers starting at  
 T0: 0  
 Number of S7 counters starting at  
 C0: 8

## Hardware interrupts

OB40 - Priority: 16

## Interrupts for DPV1

OB55 - Priority: 2  
 OB56 - Priority: 2  
 OB57 - Priority: 2

## Time-delay interrupts

OB20 - Priority: 3

## Asynchronous error interrupts

OB82 - Priority: 26  
 OB85 - Priority: 26  
 OB86 - Priority: 26  
 OB87 - Priority: 26

## Time-of-day interrupts

OB10  
 Priority: 2  
 Active: No  
 Execution: None  
 Start date: 01.01.94  
 Start time: 00:00

Cyclic interrupts

OB35  
 Priority: 12  
 Execution: 100 ms

System diagnostics

Expanded functional scope: No  
 Report cause of STOP: Yes

Block

Synchronization in the PLC  
 Synchronization mode: None  
 Time interval: None  
 Synchronization on MPI  
 Synchronization mode: None  
 Time interval: None  
 Correction factor: 0 ms

Level of protection: No protection

Mode: Test mode

Communication

Connection Resources Reserved for  
 PG communication: 1  
 OP communication: 1  
 S7 standard communication: 0  
 Max. Number of connection resources: 8

Slot 0, Slot 2, Interface X2

Port designation: DP  
 Order no.: ---  
 Designation: DP  
 Width: 1  
 PROFIBUS address: 2  
 Highest PROFIBUS address: 126  
 Data rate: 1.5 Mbps

Comment: ---

Addresses

Inputs  
 Start: 1023  
 End: 1023

Slot 0, Slot 2, Interface 2

Port designation: DI16/DO16  
 Order no.: ---  
 Designation: DI16/DO16  
 Digital channels: 16 Inputs  
 16 Outputs  
 Width: 1

Comment: ---

Addresses

Inputs  
 Start: 124  
 End: 125



puts  
 Start: 124  
 End: 125

eters

put 0  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

put 1  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

put 2  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

put 3  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

put 4  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

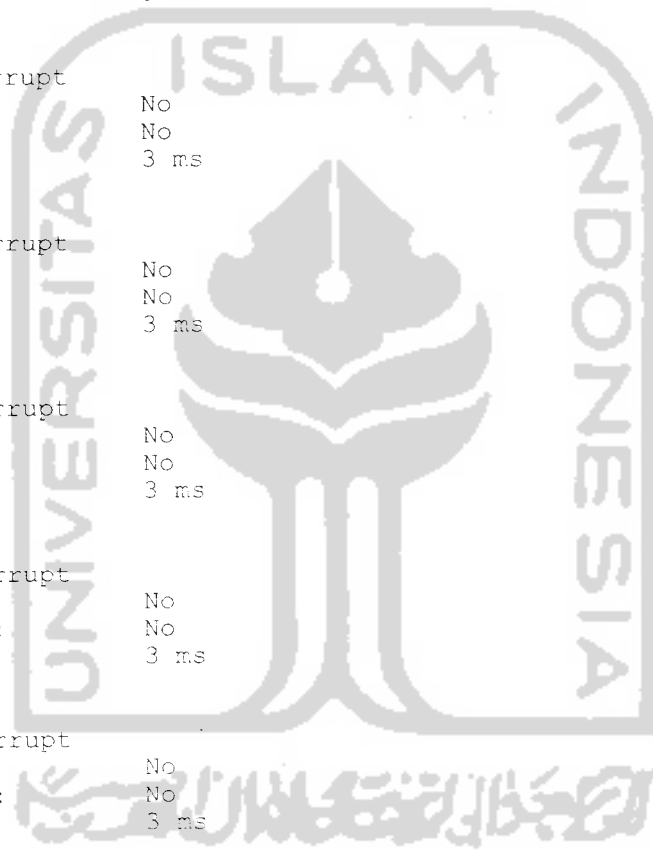
put 5  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

put 6  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

put 7  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

put 8  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

put 9  
 Trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms



t 10  
 trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

t 11  
 trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

t 12  
 trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

t 13  
 trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

t 14  
 trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

t 15  
 trigger for hardware interrupt  
 Rising (positive) edge: No  
 Falling (negative) edge: No  
 Input delay (ms): 3 ms

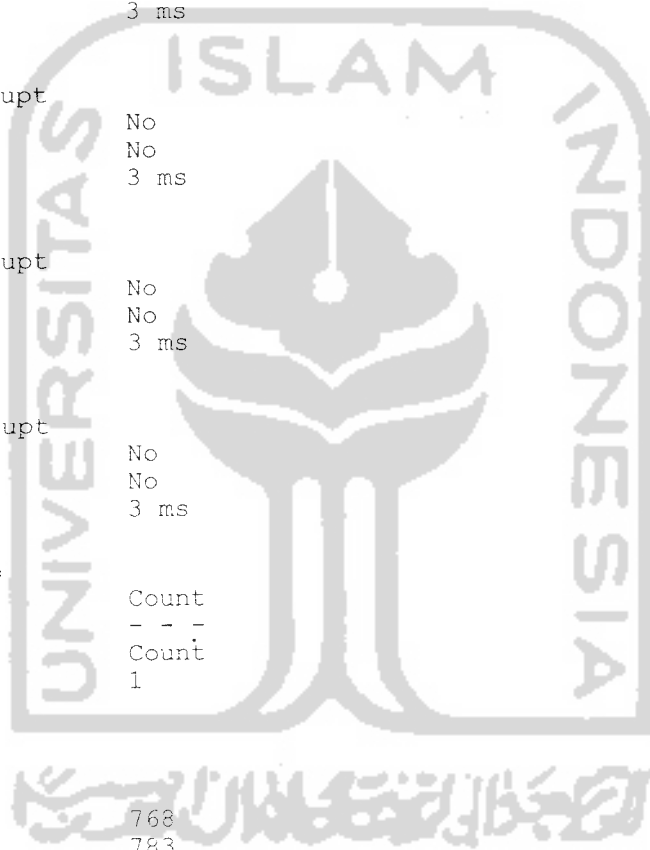
**0, Slot 2, Interface 4**

Port designation: Count  
 Order no.: - - -  
 Designation: Count  
 Width: 1  
 Comment:  
 Addresses  
 Inputs  
 Start: 768  
 End: 783  
 Outputs  
 Start: 768  
 End: 783

Parameters  
 Basic settings  
 Interrupt selection: None  
 Operating mode: Not configured

**0, Slot 4**

Port designation: DI32xDC24V  
 Order no.: 6ES7 321-1BL00-0AA0  
 Designation: DI32xDC24V  
 Digital channels: 32 Inputs  
 Width: 1  
 Comment:



```

- - -
Addresses
Inputs
  Start:      0
  End:        3
    
```

**Slot 0, Slot 5**

```

Short designation:  DO32xDC24V/0.5A
Order no.:          6ES7 322-1BL00-0AA0
Designation:        DO32xDC24V/0.5A
Digital channels:   32 Outputs
Width:              1
Comment:
- - -
    
```

```

Addresses
Outputs
  Start:      4
  End:        7
    
```

**Slot 0, Slot 6**

```

Short designation:  CP 343-1 Lean
Order no.:          6GK7 343-1CX00-0XE0
Designation:        CP 343-1 Lean
    
```

```

Location
Station:           SIMATIC 300(1)
Width:             1
    
```

```

MPI address:      3
Name of MPI network:  - - -
    
```

```

Network
Network type:     Ind. Ethernet
Network name:     Ethernet(1)
IP address:       192.168.0.2
Subnet mask:      255.255.255.0
Router address:   - - -
    
```

```

IP Configuration
Address assignment mode:  Properties- Ethernet Interface dialog
    
```

```

Addresses
Inputs
  Start:      288
  End:        16
  System selection:  - - -
Outputs
  Start:      288
  End:        16
  System selection:  - - -
    
```

```

Comment:
- - -
    
```

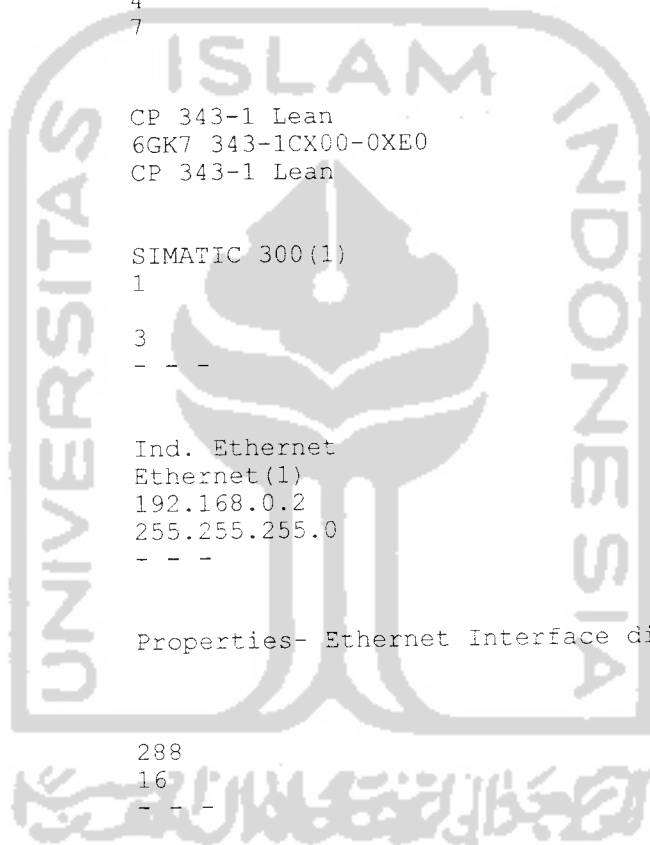
```

Time synchronization on:  No
    
```

**master system:**

```

Assigned master:
Short designation      DP
Order no.              DP
Designation            DP
Location
    
```



Station	SIMATIC 300(1)
Rack	0
Slot	2
Receptacle for interface module	1
ROFIBUS address	2

Group: 1  
 Comment:  
 The group supports SYNC.  
 The group supports FREEZE.

Group: 2  
 Comment:  
 The group supports SYNC.  
 The group supports FREEZE.

Group: 3  
 Comment:  
 The group supports SYNC.  
 The group supports FREEZE.

Group: 4  
 Comment:  
 The group supports SYNC.  
 The group supports FREEZE.

Group: 5  
 Comment:  
 The group supports SYNC.  
 The group supports FREEZE.

Group: 6  
 Comment:  
 The group supports SYNC.  
 The group supports FREEZE.

Group: 7  
 Comment:  
 The group supports SYNC.  
 The group supports FREEZE.

Group: 8  
 Comment:  
 The group supports SYNC.  
 The group supports FREEZE.



Slave : EM 277 PROFIBUS-DP PROFIBUS address: 3  
 assigned to the groups:

Slave: **EM 277 PROFIBUS-DP**

Order number: 6ES7 277-0AA2.-0XA0

```

mily: PLC
slave type: EM 277 PROFIBUS-DP
manufacturer: Siemens
D file name: SIEM089D.GSD
D revision: 1
entification number: 0x089d
vision of the DP slave: V1.02
rdware release: 1.00
ftware revision level: 1.02
nment:
OFIBUS address : 3
agnostic address: 1022
NC: Yes
EEZE: Yes
tchdog: Turned on
nchronize DP slave with DP cycle: No
rameter assignment in ascending byte sequence (hexadecimal specification) :
,13,88
    
```

```

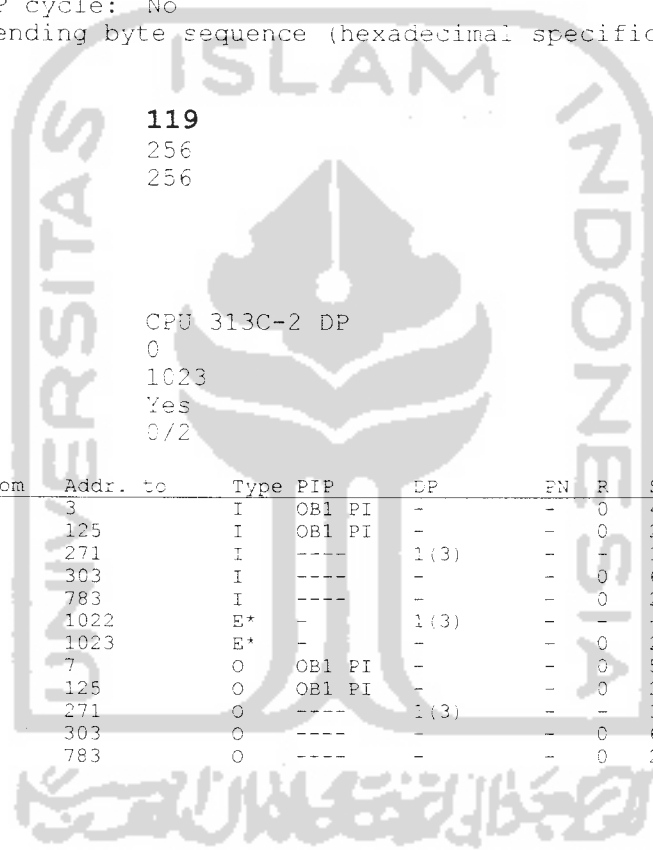
ID: 119
it address: 256
out address: 256
    
```

**ress Overview**

```

resses of: CPU 313C-2 DP
ress area from: 0
ress area to: 1023
lable address assignment: Yes
: / Slot: 0/2
    
```

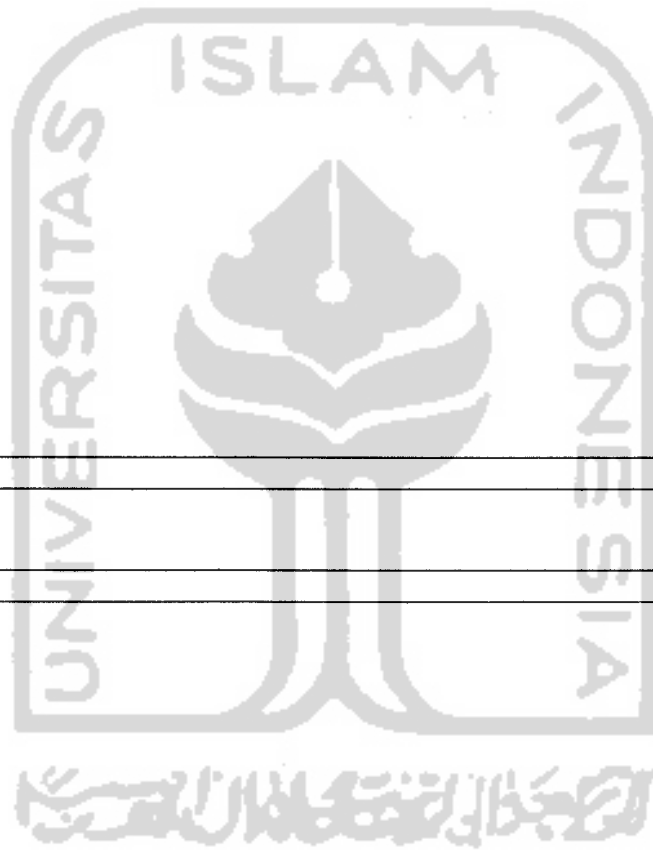
le	Addr. from	Addr. to	Type	PIP	DP	FN	R	S	IF
xDC24V	0	3	I	OB1 PI	-	-	0	4	-
/DO16	124	125	I	OB1 PI	-	-	0	2	2
	256	271	I	----	1(3)	-	-	1	-
43-1 Lean	288	303	I	----	-	-	0	6	-
t	768	783	I	----	-	-	0	2	4
77 PROFIBUS-DP	1022	1022	E*	-	1(3)	-	-	-	-
	1023	1023	E*	-	-	-	0	2	1
xDC24V/0.5A	4	7	O	OB1 PI	-	-	0	5	-
/DO16	124	125	O	OB1 PI	-	-	0	2	2
	256	271	O	----	1(3)	-	-	1	-
43-1 Lean	288	303	O	----	-	-	0	6	-
t	768	783	O	----	-	-	0	2	4



- <offline>

Family:  
 Version: 0.1  
 Block version: 2  
 tamp Code: 15/11/2006 21:13:08PM  
 Interface: 15/02/1996 16:51:12PM  
 s (block/logic/data): 00204 00058 00020

	Data Type	Address	Comment
P		0.0	
EMP0	Byte	0.0	
EMP1	Byte	1.0	
EMP2	Byte	2.0	
EMP3	Byte	3.0	
EMP4	Byte	4.0	
EMP5	Byte	5.0	
EMP6	Int	6.0	
EMP7	Int	8.0	
EMP8	Int	10.0	
EMP9	Date And Time	12.0	



OB1

rk: 1

MOVE  
 EN ENO  
 MB10 IN OUT PQB256

rk: 2

MOVE  
 EN ENO  
 MB11 IN OUT PQB271

rk: 3

MOVE  
 EN ENO  
 PIB256 IN OUT PQB4



rk: 4

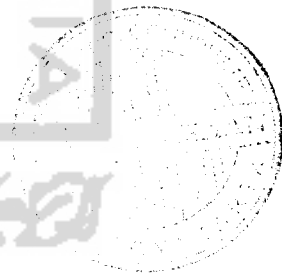
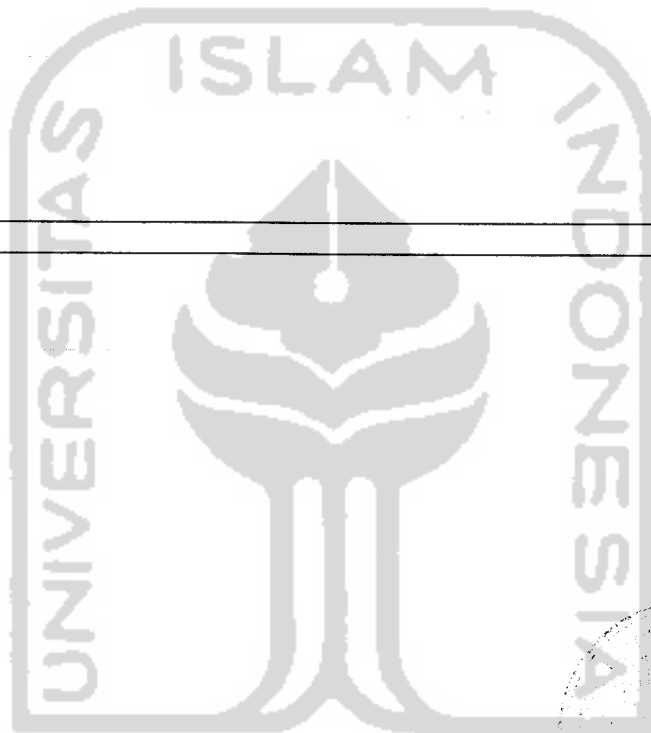
MOVE  
EN ENO  
PIB257 IN OUT PQB5

rk: 5

MOVE  
EN ENO  
PIB258 IN OUT PQB6

rk: 6

MOVE  
EN ENO  
PIB259 IN OUT PQB7



**FB1 - <offline>**

```

""
Name:
Author:
Time stamp Code.
Lengths (block/logic/data): 00114 00020 00000
Family:
Version: 0.1
Block version: 2
30/05/2005 06:31:55AM
Interface: 30/05/2005 06:26:03AM

```

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
TEMP		0.0		

Block: FB1

Network: 1

```

I0.0      I0.1      Q4.0
-----
Q4.0

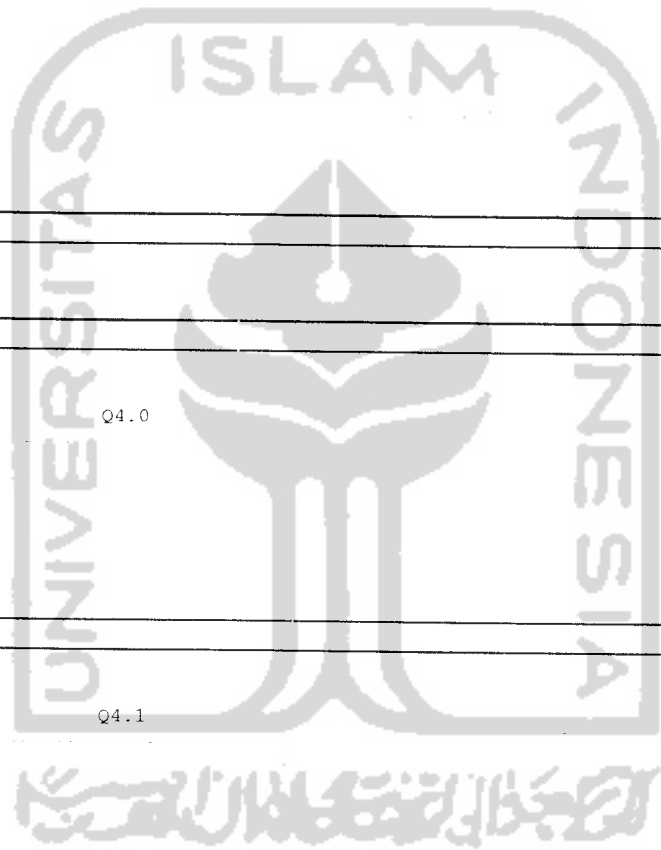
```

Network: 2

```

Q4.0      I0.2      Q4.1
-----

```



**FB10 - <offline>**

```

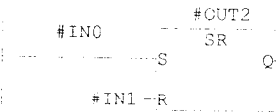
""
Name:                               Family:
Author:                              Version: 0.1
                                         Block version: 2
Time stamp Code:                     25/08/2005 14:30:25PM
                                         Interface:
                                         25/08/2005 14:30:25PM
Lengths (block/logic/data): 00118 00020 00000

```

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
IN0	Bool	0.0	FALSE	
IN1	Bool	0.1	FALSE	
OUT		0.0		
OUT2	Bool	2.0	FALSE	
IN_OUT		0.0		
STAT		0.0		
TEMP		0.0		

Block: FB10

Network: 1



FC1 - <offline>

""  
 Name: Family:  
 Author: Version: 1.1  
 Block version:  
 Time stamp Code: 07/11/2007 14:41:00  
 Interface: 07/03/2007 14:41:00  
 Lengths (block/logic/data): 00048 00100 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

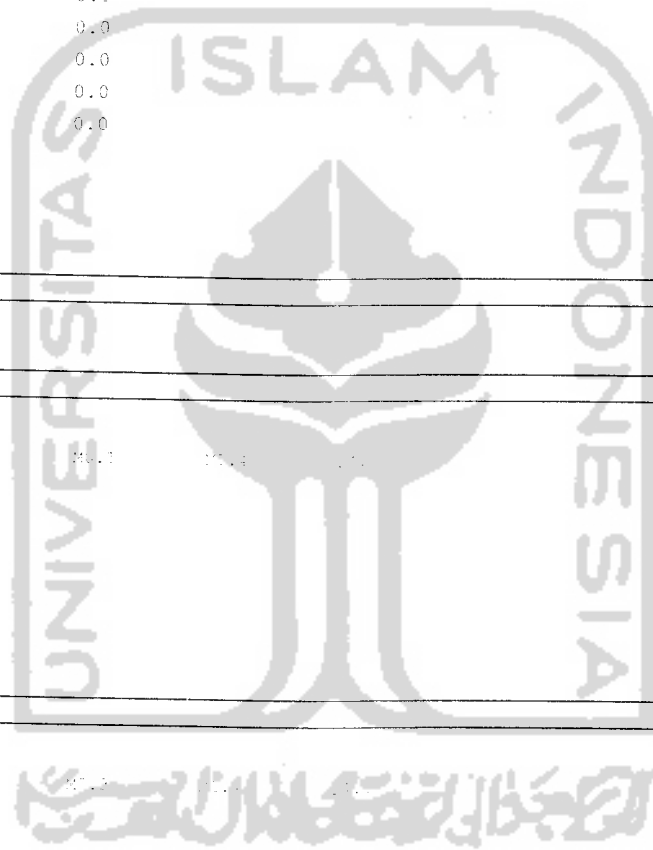
Block: FC1

Network: 1

IP	Port	Host	Protocol
192.168.1.1	1024	192.168.1.1	TCP
192.168.1.1	1024	192.168.1.1	TCP
192.168.1.1	1024	192.168.1.1	TCP

Network: 2

IP	Port	Host	Protocol
192.168.1.1	1024	192.168.1.1	TCP
192.168.1.1	1024	192.168.1.1	TCP
192.168.1.1	1024	192.168.1.1	TCP



Network: 3

10.0      10.0      255      255      255  
10.0  
20.4

Network: 4

11.1      11.1      255      255      255  
11.2  
20.6

Network: 5

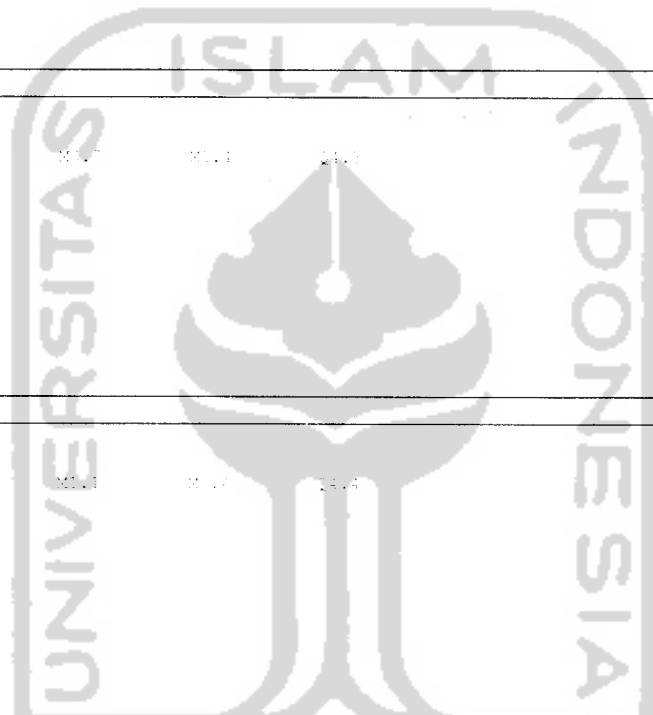
11.4      11.4      255      255      255  
11.5  
20.9

Network: 6

11.7      11.7  
10.9  
20.10

Network: 7

13.0      13.0



SIMATIC

upload irul\SIMATIC 300(1)\CPU 313C-2 DP\...\FC1 -  
<offline>

20/02/2007 14:24:2

Network: 8

13.1

20.1

Network: 9

13.2

ISLAM

Network: 10

13.3

20.2

Network: 11

13.4

20.3

Network: 12

13.5

20.4



**FC2 - <offline>**

```

""
Name:                               Family:
Author:                              Version: 0.1
Block version: 0
Time stamp Code:                     20/02/2007 14:26:51:DEM
Interface:                             20/02/2007 14:26:51:DEM
Lengths (block/logic/data):          1156 / 1750 / 1000

```

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC2

Network: 1

```

M1.1 --- 0=1 1156.1
M1.2 ---

```

Network: 1

```

M1.1 --- 0=1 1156.1
M1.2 ---

```

Network: 1

```

M1.1 --- 0=1 1156.1
M1.2 ---

```



Network: 4

M1.3 -->=1 Q135.3  
M1.3 -->=1

Network: 5

M1.4 -->=1 Q135.4  
M1.4 -->=1

Network: 6

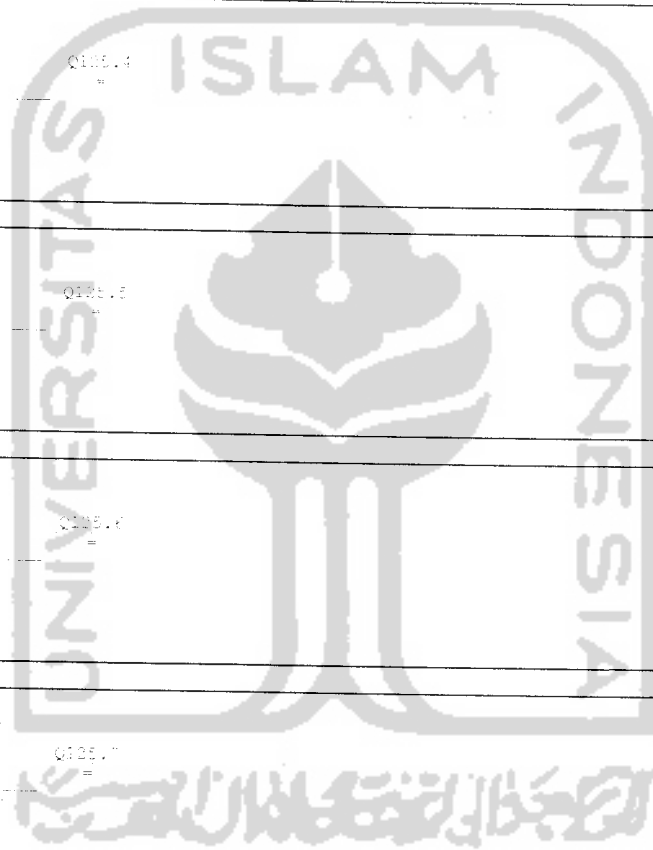
M1.5 -->=1 Q135.5  
M1.5 -->=1

Network: 7

M1.6 -->=1 Q135.6  
M1.6 -->=1

Network: 8

M1.7 -->=1 Q135.7  
M1.7 -->=1





**FC9 - <offline>**

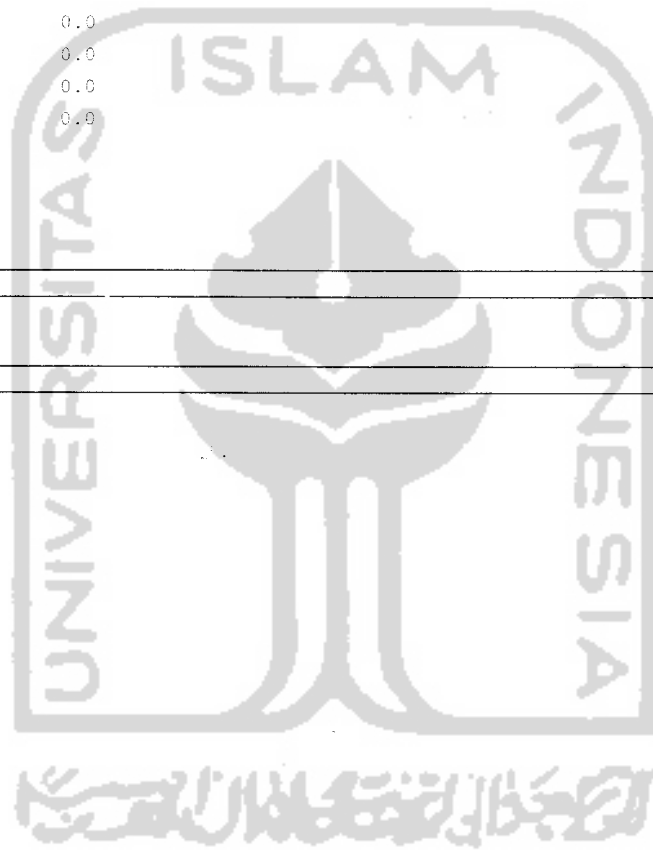
""  
Name: Family:  
Author: Version: 1.1  
Block version: 1  
Time stamp Code: 20/02/2007 14:28:43  
Interface: 300/1/1/1  
Lengths (block/logic/data): 3000/1000/1000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC9

Network: 1

Q4.0



**FC10 - <offline>**

```

""
Name: Family:
Author: Version: 0.1
Block version: 1
Time stamp Code: 25/08/2005 14:15:10PM
Interface: 25/08/2005 14:04:25PM
Lengths (block/logic/data): 00136 00040 00000

```

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC10

Network: 1

```

10.0 10.1
Q4.0

```

Network: 2

```

10.0 Q4.7 SP
10.1 R

```

Network: 3

```

EN ENO

```



10 - <offline>

Data view

block type:

Instance data block for FB10

e:

Family:

or:

Version:

0.0

Block version:

2

aths (Block / Data):

00096 / 00004

stamp

re:

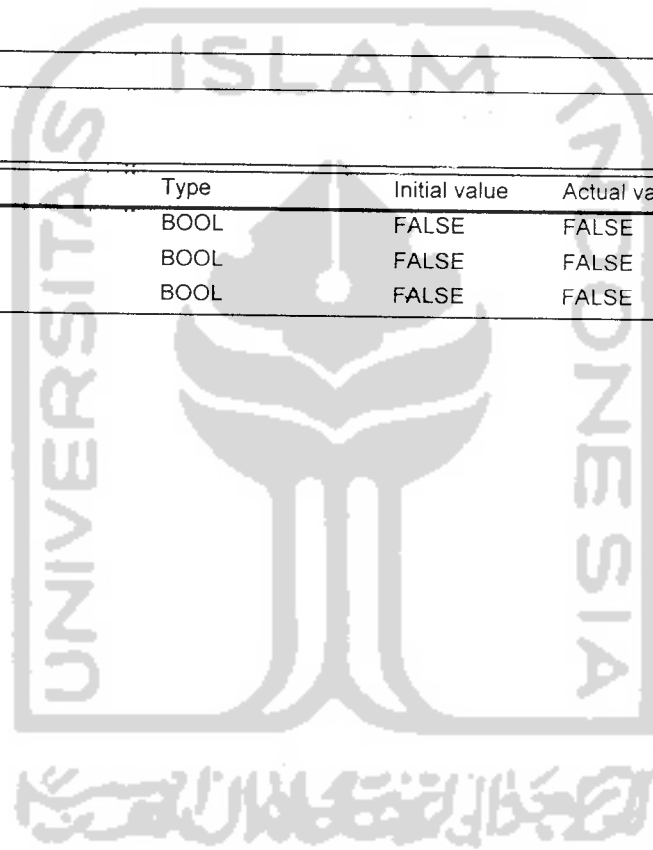
25/08/2005 14:31:01PM

face:

25/08/2005 14:30:25PM

ment:

ess	Declaration	Name	Type	Initial value	Actual value	Comment
	in	IN0	BOOL	FALSE	FALSE	
	in	IN1	BOOL	FALSE	FALSE	
	out	OUT2	BOOL	FALSE	FALSE	



upload\_irul\SIMATIC 300(1)\CPU 313C-2 DP\S7 Program(1)\Blocks

Object name	Symbolic name	Created in language	Size in the work memory	Type	Name (Header)	Version
System data	---	---	---	SDB	---	---
OB1		LAD	94	Organization Block		0.1
FB1		LAD	56	Function Block		0.1
FB10		LAD	56	Function Block		0.1
FC1		LAD	170	Function		0.1
FC2		FBD	86	Function		0.1
FC9		LAD	42	Function		0.1
FC10		LAD	76	Function		0.1
FC105		STL	244	Function	SCALE	2.1
DB10		DB	40	Instance data block for FB		0.0

WinCC™ Control Center - CS

Copyright © 1994-2004 by SIEMENS AG

\\FTI\WinCC60\_Project\_irulcoba\irulcoba.MCP

Computer		
FTI	Computer Type	Server

Tag Management

@SCRIPT_COUNT_TAGS	Data Type Group	32-Bit unsigned Script
@SCRIPT_COUNT_REQUESTS_IN_QUEUE	Parameters Data Type Group	internal tag 32-Bit unsigned Script
@SCRIPT_COUNT_ACTIONS_IN_QUEUE	Parameters Data Type Group	internal tag 32-Bit unsigned Script
@TLGRT_SIZEOF_NOTIFY_QUEUE	Parameters Data Type Group	internal tag 64-Bit IEEE 754 TagLoggingRt
@TLGRT_SIZEOF_NLL_INPUT_QUEUE	Parameters Data Type Group	internal tag 64-Bit IEEE 754 TagLoggingRt
@TLGRT_TAGS_PER_SECOND	Parameters Data Type Group	internal tag 64-Bit IEEE 754 TagLoggingRt
@TLGRT_AVERAGE_TAGS_PER_SECOND	Parameters Data Type Group	internal tag 64-Bit IEEE 754 TagLoggingRt
@CurrentUser	Parameters Data Type	internal tag Text 8-Bit
@DeltaLoaded	Parameters Data Type	internal tag 32-Bit unsigned
@LocalMachineName	Parameters Data Type	internal tag Text 8-Bit
@ConnectedRTClients	Parameters Data Type	internal tag 16-Bit unsigned
@RedundantServerState	Parameters Data Type	internal tag 16-Bit unsigned
Sensor1	Parameters Data Type Group	internal tag Bit value Conveyor
	Parameters	A4.2
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
Sensor2	Data Type Group	Bit value Conveyor
	Parameters	A4.4
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
Sensor3	Data Type Group	Bit value Conveyor
	Parameters	A4.5
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
KranBesar	Data Type Group	Bit value Conveyor
	Parameters	A6.1
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP

## Tag Management

KranKecil	Data Type	Bit value
	Group	Conveyor
	Parameters	A6.0
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
Conveyor1	Channel unit	TCP/IP
	Data Type	Bit value
	Group	Conveyor
	Parameters	A6.2
	Connection	PLC
Conveyor2	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
	Data Type	Bit value
	Group	Conveyor
	Parameters	A6.3
PosTutup	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
	Data Type	Bit value
	Group	Conveyor
TekanTutup	Parameters	A7.1
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
	Data Type	Bit value
Sensor4	Group	Conveyor
	Parameters	A4.7
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
DorongKaleng	Data Type	Bit value
	Group	Conveyor
	Parameters	A7.0
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
test	Channel unit	TCP/IP
	Data Type	Bit value
	Parameters	internal tag
	Data Type	Bit value
	Group	Conveyor
Start	Parameters	M10.0
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
	Data Type	Bit value
Stop	Group	Conveyor
	Parameters	M10.1
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
RunStop	Data Type	Bit value
	Group	Conveyor
	Parameters	M0.0

WinCC™ Control Center - CS

Copyright © 1994-2004 by SIEMENS AG

\\FTIWinCC60\_Project\_irulcobalirulcoba.MCP

Tag Management

	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
Timinglsi	Data Type	16-Bit unsigned
	Group	Conveyor
	Parameters	MW10
	Low limit	0
	High limit	2000
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
TimingRefill	Data Type	16-Bit unsigned
	Group	Conveyor
	Parameters	MW12
	Low limit	0
	High limit	2000
	Connection	PLC
	Channel	SIMATIC S7 PROTOCOL SUITE
	Channel unit	TCP/IP
Level	Data Type	16-Bit unsigned
	Parameters	internal tag
STAR	Data Type	16-Bit unsigned
	Parameters	internal tag
STOP_1	Data Type	16-Bit unsigned
	Parameters	internal tag
Connections		
PLC	Unit	TCP/IP
	Parameters	IP.192.168.0.2,,0,2,02

