



UNIVERSITAS INDONESIA

Perancangan Sistem SCADA (*Supervisory Control and Data Acquisition*) Pada Miniatur Warehouse Berbasis PLC (*Programmable Logic Controller*)

SKRIPSI

**BACHTIAR SALIM WINARNO
0806365526**

**FAKULTAS TEKNIK
PROGRAM SARJANA EKSTENSI
DEPOK
JUNI 2010**



UNIVERSITAS INDONESIA

Perancangan Sistem SCADA (*Supervisory Control and Data Acquisition*) Pada Miniatur *Warehouse* Berbasis PLC (*Programmable Logic Controller*)

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**BACHTIAR SALIM WINARNO
0806365526**

**FAKULTAS TEKNIK
PROGRAM SARJANA EKSTENSI
DEPOK
JUNI 2010**

LEMBAR PERNYATAAN ORISINALITAS

Skripsi adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

NAMA : BACHTIAR SALIM WINARNO

NPM : 0806365526

TANDA TANGAN :

TANGGAL : 15 Juni 2010

LEMBAR PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Bachtiar Salim Winarno
NPM : 0806365526
Program Studi : Teknik Elektro
Judul Skripsi : Perancangan Sistem SCADA (*Supervisory Control and Data Acquisition*) Pada Miniatur Warehouse Berbasis PLC (*Programmable Logic Controller*)

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Budi Sudiarto ST,MT (.....)
Penguji : Prof. Dr. Ir. Iwa Garniwa M K, MT (.....)
Penguji : Ir. Amien Rahardjo MT (.....)

Ditetapkan di : Depok
Tanggal : 1 Juli 2010

KATA PENGANTAR

Puji dan syukur saya panjatkan kehadirat Allah SWT, yang dengan rahmat-Nya penulis dapat menyelesaikan tugas akhir ini dengan baik. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektro pada Fakultas Teknik Universitas Indonesia. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Bapak Budi Sudiarto ST, MT selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan tugas akhir ini;
- (2) Prof. Dr. Ir. Iwa Garniwa M.K. MT, yang telah membantu memberikan ide dan masukan dalam perancangan alat;
- (3) Keluarga penulis yang telah memberikan bantuan dukungan material dan moral; dan
- (4) Sahabat yang telah banyak membantu penulis dalam menyelesaikan tugas akhir ini.

Akhir kata, Semoga Allah SWT berkenan membalas segala kebaikan semua pihak yang telah banyak membantu. Semoga tugas akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 14 Juni 2010

Penulis

Bachtiar Salim Winarno

NPM. 0806365526

**LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan dibawah ini :

Nama : Bachtiar Salim Winarno

NPM : 0806365526

Program Studi : Teknik Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Perancangan Sistem SCADA (*Supervisory Control and Data Acquisition*) Pada Miniatur Warehouse Berbasis PLC (*Programmable Logic Controller*)

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai Hak Cipta.

Demikian pernyataan ini saya buat sebenarnya.

Dibuat di : Depok

Pada Tanggal : 15 Juni 2010

Yang menyatakan

(Bachtiar Salim Winarno)

ABSTRAK

Nama : Bachtiar Salim Winarno
Program Studi : Teknik Elektro
Judul : Perancangan Sistem SCADA (*Supervisory Control and Data Acquisition*) Pada Miniatur Warehouse Berbasis PLC (*Programmable Logic Controller*)

Skripsi ini membahas tentang perancangan sistem SCADA (*Supervisory Control and Data Acquisition*) pada sebuah miniatur warehouse. Pada perancangan sistem ini digunakan PLC (*Programmable Logic Controller*) Omron CJ1M sebagai RTU (*Remote Terminal Unit*) yang mengambil data sensor dan unit-unit beban kemudian menyampaikannya pada MTU (*Master Terminal Unit*) yang terdiri dari HMI (*Human Machine Interface*) SCADA. Pada skripsi ini dirancang sistem SCADA sederhana untuk mengawasi dan mengontrol unit-unit beban dan sensor yang ada dalam miniatur warehouse, yaitu lampu dan fan serta sensor *smoke detector* dan *motion sensor*. Kemudian untuk HMI SCADA akan dibuat dengan bahasa pemrograman *Visual Basic 6.0* dengan *Microsoft Access 2007* sebagai media penyimpanan *database*-nya.

Pada perancangan HMI SCADA terdapat beberapa mode yaitu *local* dan *remote*. Mode *local* digunakan untuk mengontrol unit-unit beban dari miniatur warehouse dan mode *remote* digunakan untuk mengontrol unit-unit beban dari HMI SCADA / Laptop. Untuk pengujian sistem dilakukan pengukuran waktu respon dari HMI SCADA terhadap miniatur warehouse dan begitu juga sebaliknya. Dari pengujian tersebut dapat disimpulkan bahwa pada mode *local* waktu respon terhadap miniatur warehouse lebih cepat dibandingkan mode *remote* dan waktu respon terhadap HMI SCADA lebih lama dibanding mode *remote*. Lalu untuk mode *remote* waktu respon terhadap miniatur warehouse dan HMI SCADA hampir sama.

KATA KUNCI :

SCADA, PLC, *Visual Basic*, Warehouse.

ABSTRACT

Name : Bachtiar Salim Winarno
Study Program : Electrical Engineering
Title : Design of SCADA (Supervisory Control and Data Acquisition)
System on Warehouse Miniature Based on PLC (Programmable Logic Controller)

This thesis discusses the design of SCADA systems (Supervisory Control and Data Acquisition) on a miniature warehouse. In designing this system is used PLC (Programmable Logic Controller) Omron CJ1M as RTU (Remote Terminal Unit) which takes sensor data and load units then pass them on to the MTU (Master Terminal Unit) consisting of HMI (Human Machine Interface), SCADA. In this paper designed a simple SCADA system for monitoring and control units and sensors of existing loads in the miniature of warehouse, which is light and fan and smoke detector sensors and motion sensors. Then for SCADA HMI will be created with Visual Basic 6.0 with Microsoft Access 2007 as database storage media.

In SCADA HMI design, there are several modes of local and remote. Local mode is used to control the load units from the warehouse miniature and the remote mode is used to control the load units from SCADA HMI / Laptop. To test the system response time measurements from SCADA HMI on the warehouse miniature and vice versa. From these tests we can conclude that the local modes of warehouse miniature response time is faster than remote mode and time of the response to be longer than SCADA HMI remote mode. Then for remote mode response time to warehouse miniature and SCADA HMI is almost the same.

KEYWORD :

SCADA, PLC, Visual Basic, Warehouse.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERNYATAAN ORISINALITAS	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB 1 PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Metode Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB 2 LANDASAN TEORI	
2.1. SCADA (<i>Supervisory Control and Data Acquisition</i>).....	5
2.2. <i>Microsoft Visual Basic 6.0</i>	9
2.2.1. <i>Object Timer</i>	13
2.2.2. <i>Object Common Dialog</i>	14

2.2.3. Tipe Data	15
2.2.4. Struktur Keputusan <i>If Then</i>	17
2.2.5. <i>Loop For Next</i>	17
2.3. <i>Microsoft Access 2007</i>	17
2.4. PLC (<i>Programmable Logic Controller</i>).....	19
2.4.1 PLC Omron CJ1M-CPU11	20
2.4.2 PLC Input CJ1W-ID211	21
2.4.3 PLC Output CJ1W-OD211	21
2.4.4 PLC Power Supply CJ1W-PA202	22
2.5. Komunikasi <i>Microsoft Visual Basic 6.0</i> dengan PLC.....	22
2.6. CX-Programmer 7.0	23
2.7. Detektor Asap (<i>Smoke Detector</i>).....	25
2.8. Sensor Gerak (<i>Motion Sensor</i>)	25

BAB 3 PERANCANGAN SISTEM

3.1 Cara Kerja Sistem	27
3.2 Perancangan Miniatur <i>Warehouse</i>	28
3.3 Perancangan <i>Software</i>	30
3.3.1 Perancangan <i>Ladder Diagram</i>	31
3.3.2 Perancangan <i>Software HMI (Human Machine Interface)</i>	41
3.3.2.1 <i>Form Loading</i>	41
3.3.2.2 <i>Form Main Program</i>	42
3.3.2.3 <i>Form Time Usage</i>	43
3.3.2.4 <i>Form History Graph</i>	44
3.3.2.5 <i>Form Setting</i>	45
3.3.3 Koneksi Database.....	46

BAB 4 PENGUJIAN SISTEM

4.1 Pengujian mode <i>Local</i>	47
4.2 Pengujian mode <i>Remote</i>	48
4.3 Pengujian Program <i>Smoke Detector</i>	49
4.4 Pengujian Program <i>Motion Sensor</i>	50

4.5 Pengujian Program <i>Reset Alarm</i>	51
4.6 Pengujian Program <i>Reset System</i>	52
4.7 Pengujian Program <i>Emergency</i>	53
4.8 Pengujian Program <i>Hours Meter</i>	53

BAB 5 KESIMPULAN	56
-------------------------------	-----------

DAFTAR REFERENSI	57
-------------------------------	-----------

DAFTAR GAMBAR

2.1	Bagian sistem SCADA sederhana	5
2.2	Lingkungan dalam pemrograman <i>Visual Basic 6.0</i>	10
2.3	Toolbar	11
2.4	Toolbox.....	11
2.5	<i>Form Window</i>	12
2.6	<i>Project Explorer</i>	12
2.7	<i>Properties Window</i>	13
2.8	<i>Object Timer</i>	14
2.9	<i>Properties Object Timer</i>	14
2.10	<i>Object Common Dialog</i>	15
2.11	CJ1M-CPU11	21
2.12	PLC <i>Input</i> CJ1W-ID211	21
2.13	PLC <i>Output</i> CJ1W-OD211	22
2.14	PLC <i>Power Supply</i> CJ1W-PA202	22
2.15	Kontak NO.....	23
2.16	Kontak NC.....	23
2.17	<i>Coil</i>	23
2.18	<i>Timer</i>	24
2.19	<i>MOV</i>	24
2.20	<i>Smoke Detector</i>	25
2.21	<i>Motion Sensor</i>	26
3.1	Blok diagram cara kerja sistem	27
3.2	Desain alas <i>warehouse</i>	28
3.3	Desain bangunan <i>warehouse</i>	29
3.4	Desain keseluruhan <i>warehouse</i>	29
3.5	<i>Flowchart Local System</i>	33
3.6	<i>Flowchart Local System (Lanjutan)</i>	34

3.7	<i>Flowchart Remote System</i>	35
3.8	<i>Flowchart Remote System (Lanjutan)</i>	36
3.9	<i>Ladder diagram mode local dan remote</i>	37
3.10	<i>Flowchart untuk sistem gangguan</i>	38
3.11	<i>Ladder diagram program alarm</i>	39
3.12	<i>Ladder diagram program alarm (Lanjutan)</i>	40
3.13	<i>Form Loading</i>	41
3.14	<i>Form Main Program</i>	42
3.15	<i>Form Time Usage</i>	43
3.16	<i>Form History Graph 1</i>	44
3.17	<i>Form History Graph 2</i>	45
3.18	<i>Form Setting Load</i>	46
3.19	<i>ADODC</i>	46

DAFTAR TABEL

2.1 Pilihan pada <i>common dialog</i>	15
2.2 Tipe Data.....	16
3.1 <i>Input</i> PLC CJ1W-ID211 yang digunakan.....	31
3.2 <i>Output</i> PLC CJ1W-OD211 yang digunakan.....	32
4.1 Hasil pengujian 1 mode <i>local</i>	48
4.2 Hasil pengujian 2 mode <i>local</i>	48
4.3 Hasil pengujian 1 mode <i>remote</i>	49
4.4 Hasil pengujian 2 mode <i>remote</i>	49
4.5 Hasil pengujian 1 program <i>smoke detector</i>	50
4.6 Hasil pengujian 2 program <i>smoke detector</i>	50
4.7 Hasil pengujian 1 program <i>motion sensor</i>	51
4.8 Hasil pengujian 2 program <i>motion sensor</i>	51
4.9 Hasil pengujian 1 program <i>reset alarm</i>	51
4.10 Hasil pengujian 2 program <i>reset alarm</i>	52
4.11 Hasil pengujian 1 program <i>reset system</i>	52
4.12 Hasil pengujian 2 program <i>reset system</i>	52
4.13 Hasil pengujian 1 program <i>emergency</i>	53
4.14 Hasil pengujian 2 program <i>emergency</i>	53
4.15 Hasil pengujian 1 program <i>hours meter</i>	54
4.16 Hasil pengujian 2 program <i>hours meter</i>	54

DAFTAR LAMPIRAN

Lampiran 1. Diagram Rangkaian dan Gambar Miniatur

Lampiran 2. *Source Code* Program HMI SCADA

Lampiran 3. *Ladder Diagram* PLC Omron CJ1M

Lampiran 4. Spesifikasi Teknis PLC Omron CJ1M

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam perkembangan teknologi yang begitu pesat dibutuhkan kecepatan dan efisiensi dalam pengontrolan alat atau mesin, serta dibutuhkan pengolahan data untuk laporan. Dengan cara pengontrolan sistem konvensional, beberapa industri semakin tertinggal dalam persaingan baik dalam kualitas dan kuantitas produk. Oleh sebab itu sekarang ini semakin banyak perusahaan dan industri yang menggunakan sistem berbasis SCADA (*Supervisory Control and Data Acquisition*) untuk mengawasi dan mengontrol unit-unit beban yang ada di *plant* serta dapat mengakuisisi data yang terdapat pada *plant*. Oleh karena itu dalam tugas akhir ini akan dibuat perancangan sistem SCADA untuk mengawasi dan mengontrol unit-unit beban pada miniatur *warehouse* yaitu berupa pengontrolan untuk unit penerangan (Lampu), pendingin ruangan (*Fan*), dan sensor pendeteksi kebakaran (*Smoke Detector*) serta dilengkapi dengan sensor gerak (*Motion Sensor*) untuk sistem keamanan.

Penerapan sistem SCADA ini dimaksudkan untuk mengawasi dan mengontrol penggunaan unit-unit beban pada miniatur *warehouse* agar dapat dilakukan efisiensi dalam pemakaian unit-unit beban tersebut. Dalam sistem ini akan dilakukan akuisisi data yaitu untuk aktifitas pada unit-unit penerangan (Lampu) dan pendingin ruangan (*Fan*) serta gangguan yang terjadi pada miniatur *warehouse*. Data tersebut diolah untuk mengetahui lama pemakaian dari masing-masing unit penerangan dan pendingin ruangan. Untuk mengontrol semua itu maka digunakan PLC (*Programmable Logic Controller*) sebagai RTU (*Remote Terminal Unit*), PLC ini juga yang digunakan untuk menerima masukan dari saklar dan sensor-sensor.

1.2 Perumusan Masalah

Permasalahan yang akan dibahas disini adalah perancangan sistem SCADA pada miniatur *warehouse*. Perancangan sistem SCADA ini terdiri dari perancangan *software* HMI (*Human Machine Interface*) SCADA dengan bahasa pemrograman *Visual Basic 6.0* dan perancangan *ladder diagram* PLC dengan *CX-Programmer 7.0* Kemudian akan dibahas mengenai bagaimana menghubungkan *Visual Basic 6.0* dengan *Microsoft Access 2007* sebagai media penyimpanan *database*.

1.3 Pembatasan Masalah

Adapun hal-hal yang membatasi tugas akhir ini adalah:

1. Unit-unit beban pada miniatur *warehouse* yang diawasi dan dikontrol secara *real time* hanya unit penerangan (Lampu) dan pendingin ruangan (*Fan*).
2. Lampu dan *Fan* yang diawasi dan dikontrol hanya beberapa saja sebagai simulasi miniatur *warehouse*.
3. Sensor yang diawasi dan dikontrol yaitu sensor pendeteksi asap (*Smoke Detector*) dan sensor gerak (*Motion Sensor*). Pembahasan tentang sensor tidak dibahas secara lengkap.
4. Sistem akuisisi datanya hanya berupa aktivitas dari pengoperasian lampu dan *fan* untuk digunakan dalam menghitung lama pemakaian unit-unit beban tersebut.
5. Perancangan *software* HMI SCADA menggunakan *Microsoft Visual Basic 6.0*.

1.4 Tujuan

Adapun tujuan yang ingin dicapai dalam tugas akhir ini adalah:

1. Mengimplementasikan sistem SCADA pada sebuah miniatur *warehouse*.
2. Untuk pembelajaran dalam perancangan sistem SCADA secara keseluruhan.

1.5 Metodologi Penelitian

Dalam tugas akhir ini digunakan beberapa metodologi yaitu :

1. Studi Literatur
Mencari dan mempelajari dasar-dasar teori yang berkaitan dengan topik dari buku ataupun media lain yang mendukung. Sesuai yang akan dibahas yaitu sistem SCADA, PLC, *Visual Basic 6.0* dan komponen lain yang akan dipakai.
2. Studi Analisis
Penganalisaan sistem dan program dalam perancangan maupun pengujian akhir.
3. Studi Aplikatif
Mengaplikasikan langsung teori-teori yang bersangkutan dalam miniatur alat dan sistem yang akan dirancang.

1.6 Sistematika Penulisan

Untuk mempermudah pemahaman terhadap permasalahan yang dibahas dalam skripsi ini laporan skripsi ini menggunakan sistematika penulisan yaitu bab 1 pendahuluan yang berisi tentang latar belakang, perumusan masalah, pembatasan masalah, tujuan, metodologi penulisan dan sistematika penulisan. Kemudian bab 2 landasan teori berisi tentang teori-teori dasar yang menunjang perancangan sistem seperti dasar teori SCADA, PLC dan *Visual Basic 6.0*. Lalu bab 3 perancangan sistem membahas tentang perancangan sistem mulai dari desain miniatur *warehouse*, perancangan *ladder diagram* PLC dan *software* untuk HMI SCADA. Pada bab 4 pengujian sistem membahas metode dan hasil pengujian pada HMI SCADA yang telah dirancang. Kemudian pada bab 5 kesimpulan berisi mengenai kesimpulan dari perancangan dan pengujian sistem.

BAB 2

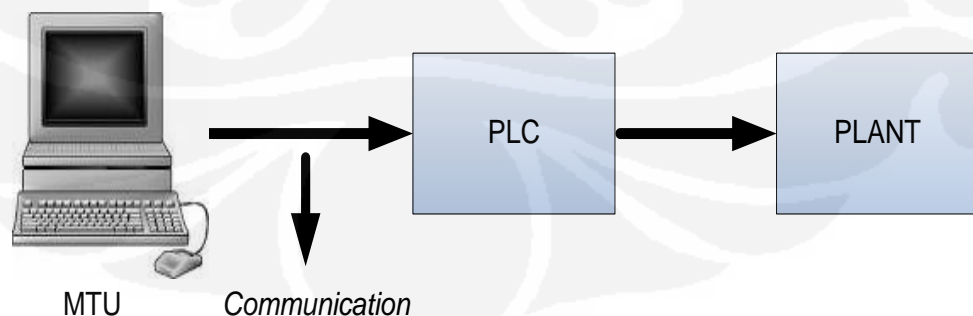
LANDASAN TEORI

2.1 SCADA

SCADA (*Supervisory Control and Data Acquisition*) adalah suatu sistem yang dapat melakukan pengawasan, pengendalian dan akuisisi data terhadap sebuah *plant*. Seiring dengan perkembangan komputer yang pesat beberapa dekade terakhir, maka komputer menjadi komponen penting dalam sebuah sistem SCADA modern. Sistem ini menggunakan komputer untuk menampilkan status dari sensor dan aktuator dalam suatu *plant*, menampilkannya dalam bentuk grafik dan menyimpannya dalam *database*. Umumnya komputer ini terhubung dengan sebuah pengendali (misal : PLC) melalui sebuah protokol komunikasi tertentu (misal : *serial communication*).

Bagian – bagian SCADA yaitu :

1. Sensor dan aktuator (*Field Device*)
2. *Remote Terminal Unit / PLC (Programmable Logic Controller)*
3. *Sistem Komunikasi*
4. *Master Terminal Unit*



Gambar 2.1 Bagian sistem SCADA sederhana

Berikut ini penjelasan dari masing-masing bagian :

1. Sensor dan aktuator (*Field device*)

Bagian ini adalah *plant* yang ada di lapangan yang terdiri dari obyek yang memiliki berbagai sensor dan aktuator. Sensor merupakan sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang dirubah menjadi besaran listrik disebut *transduser*. Nilai sensor dan aktuator inilah yang umumnya diawasi dan dikendalikan supaya obyek/*plant* berjalan sesuai dengan keinginan pengguna.

2. Remote Terminal Unit (RTU)

RTU Merupakan unit-unit komputer kecil (mini), maksudnya sebuah unit yang dilengkapi dengan sistem mandiri seperti sebuah komputer, yang ditempatkan pada lokasi dan tempat-tempat tertentu di lapangan. RTU bertindak sebagai pengumpul data lokal yang mendapatkan datanya dari sensor-sensor dan mengirimkan perintah langsung ke peralatan di lapangan. Pada sistem SCADA, RTU berbeda dari *Programmable Logic Controller* (PLC) dalam RTU lebih cocok untuk telemetri geografis yang luas, yang sering menggunakan komunikasi nirkabel, sementara PLC lebih cocok untuk daerah kontrol (*plant*, jalur produksi, dll) di mana sistem menggunakan media fisik untuk kontrol. Dalam sistem *basic* SCADA umumnya digunakan PLC sebagai pengganti RTU.

3. Sistem Komunikasi

Sistem komunikasi SCADA diperlukan untuk menghubungkan antara MTU dengan PLC. Pada awalnya komunikasi data melalui radio, modem atau jalur kabel serial khusus. Saat ini data-data SCADA dapat disalurkan melalui jaringan Ethernet atau TCI/IP. Komunikasi SCADA diatur melalui suatu protokol

Pada umumnya sensor dan *relay* kontrol hanyalah peralatan listrik yang sederhana, alat-alat tersebut tidak bisa menghasilkan atau menterjemahkan protokol komunikasi. Dengan demikian dibutuhkan PLC

yang menghubungkan antara sensor dan jaringan SCADA. PLC mengubah masukan-masukan sensor ke format protokol yang bersangkutan dan mengirim ke master SCADA. Selain itu PLC juga menerima perintah dalam format protokol dan memberikan sinyal listrik yang sesuai ke *relay* kontrol yang bersangkutan.

Berikut ini beberapa sistem komunikasi yang dipakai dalam sistem SCADA :

- RS-232
- *Private Network* (LAN/RS-485)
- Internet
- *Wireless Communication systems*
 - *Wireless LAN*
 - *GSM Network*
 - *Radio modems*

4. MTU – SCADA Software

Master Terminal Unit umumnya ialah komputer yang memiliki SCADA *software*. Fitur – fitur yang umumnya ada pada suatu SCADA Software ialah :

- *Human Machine Interface* (HMI)
Tampilan yang memudahkan manusia (operator) untuk memahami atau mengendalikan sistem atau *plant*.
- *Graphic Displays*
Tampilan grafis, bukan hanya angka, untuk mempermudah pengamatan.
- *Alarms*
Alarm untuk memberi peringatan saat terjadi gangguan.
- *History Graph*
Grafik yang menampilkan data pengolahan pada sistem SCADA.
- *RTU / PLC Interface*
Bagian program yang menghubungkan PLC dengan SCADA *software*.

- *Database*

Penyimpanan data ke dalam *database*. (Handy Wicaksono, 2009)

SCADA bukanlah teknologi khusus tapi lebih merupakan aplikasi. Semua aplikasi yang mendapatkan data-data suatu sistem di lapangan dengan tujuan pengontrolan sistem merupakan sebuah aplikasi SCADA. Ada dua elemen dalam aplikasi SCADA yaitu :

1. Proses, sistem, mesin yang akan dipantau dan dikontrol bisa berupa *power plant*, sistem pengairan, jaringan komputer, sistem lampu trafik lalu lintas atau apa saja.
2. Sebuah jaringan peralatan cerdas dengan antarmuka ke sistem melalui sensor dan luaran kontrol. Dengan jaringan ini, yang merupakan sistem SCADA membolehkan untuk melakukan pengawasan dan pengontrolan komponen-komponen tersebut.

Sebagai contoh, SCADA digunakan di seluruh dunia antara lain untuk:

- Penghasil, transmisi dan distribusi listrik : Dalam hal ini SCADA digunakan untuk mendeteksi besarnya arus dan tegangan, pengawasan operasional *circuit breaker* dan untuk mematikan / menghidupkan *the power grid*;
- Penampungan dan distribusi air : Dalam hal ini SCADA digunakan untuk memantau dan pengaturan laju aliran air, tinggi reservoir, tekanan pipa dan berbagai macam faktor lainnya;
- Bangunan, fasilitas dan lingkungan : Dalam hal ini SCADA digunakan untuk mengontrol HVAC, unit-unit pendingin, penerangan dan sistem keamanan.
- Produksi : Aplikasi SCADA digunakan untuk mengatur inventori komponen-komponen, mengatur otomasi alat atau robot, memantau proses dan kontrol kualitas.
- Transportasi KA listrik : Aplikasi SCADA dapat digunakan untuk pemantauan dan pengontrolan distribusi listrik, otomasi sinyal

trafik KA, melacak dan menemukan lokasi KA, mengontrol palang KA dan lain sebagainya;

- Lampu lalu-lintas : Aplikasi SCADA dapat digunakan untuk memantau lampu lalu-lintas, mengontrol laju trafik, dan mendeteksi sinyal-sinyal yang salah.

Dan tentunya masih banyak lagi aplikasi-aplikasi potensial untuk sistem SCADA. SCADA saat ini digunakan hampir di seluruh industri dan infrastruktur umum. Pada dasarnya SCADA dapat digunakan dalam aplikasi-aplikasi yang membutuhkan kemudahan dalam pemantauan sekaligus juga pengontrolan dengan berbagai macam media antarmuka dan komunikasi yang tersedia saat ini misalnya komputer, PDA, *touch screen*, TCP/IP, *wireless* dan sebagainya.

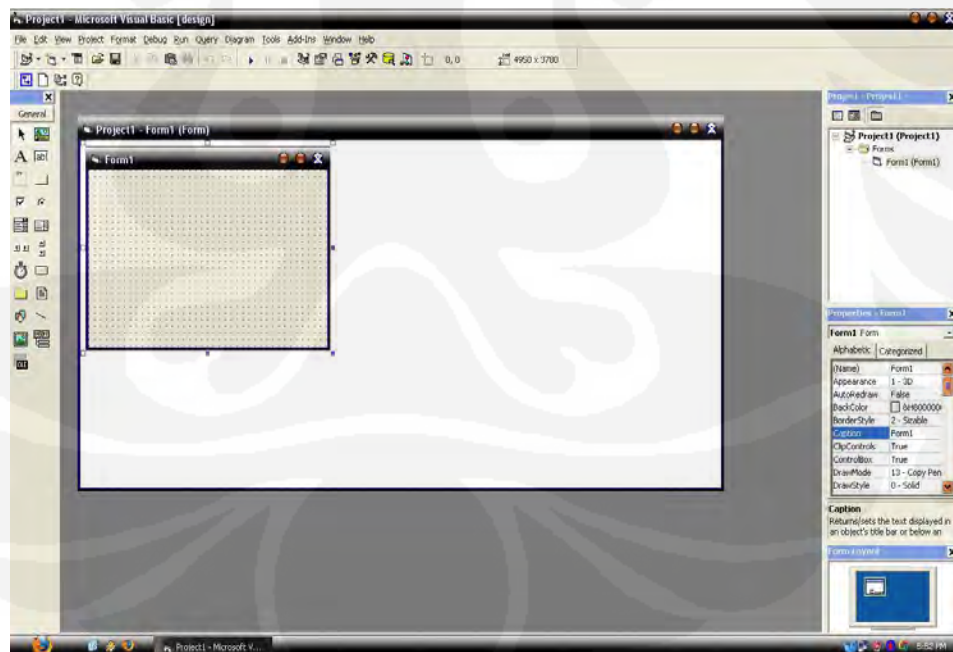
Untuk akuisisi data pada SCADA, data yang didapat berasal dari sensor-sensor yang terdapat di *plant*. Pada sistem SCADA yang kompleks mungkin dibutuhkan pemantauan terhadap ratusan hingga ribuan sensor yang tersebar di seluruh area yang terdiri dari beberapa *plant*.

Beberapa sensor dapat melakukan pengukuran kejadian secara sederhana yang bisa dideteksi menggunakan saklar ON/OFF, masukan seperti ini disebut sebagai **masukan diskrit** atau **masukan digital**. Misalnya untuk mengetahui apakah sebuah alat sudah bekerja (ON) atau belum (OFF), konveyornya sudah jalan (ON) atau belum (OFF), mesinnya sudah mengaduk (ON) atau belum (OFF), dan lain sebagainya. Beberapa sensor yang lain bisa melakukan pengukuran secara kompleks, dimana angka atau nilai tertentu itu sangat penting, masukan seperti ini disebut **masukan analog**, bisa digunakan untuk mendeteksi perubahan secara kontinu pada tegangan, arus, suhu, dan lain sebagainya. (Agfianto Eko Putra, 2009, 21, chap 1)

2.2 Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 merupakan bahasa pemrograman visual yang digunakan secara umum untuk membuat program aplikasi yang bekerja menggunakan *Operating System (OS) Microsoft Windows*. Bahasa pemrograman *Visual Basic* yang dikembangkan oleh *Microsoft* sejak tahun 1991, merupakan pengembangan dari bahasa pemrograman pendahulunya, yaitu bahasa pemrograman *BASIC (Beginner's All-purpose Symbolic Instruction Code)*, yang dikembangkan pada era 1950-an. *Visual Basic* merupakan salah satu bahasa pemrograman komputer yang mendukung *OOP (Object Oriented Programming)*. *Visual Basic 6.0* cukup *user-friendly*, dalam arti pengguna pasti akan merasakan mudahnya membuat program aplikasi dengan tampilan grafis yang cukup baik dalam waktu yang relatif singkat.

Dalam lingkungan pemrograman *Visual Basic 6.0* terdapat komponen-komponen yang biasa digunakan yaitu :



Gambar 2.2 Lingkungan dalam pemrograman *Visual Basic 6.0*

1. *Toolbar*

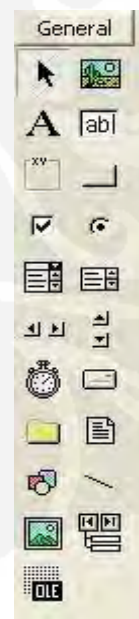
Toolbar merupakan sebuah komponen yang berisi kumpulan tombol yang terletak dibagian bawah menu bar. Pada kondisi default program *Visual Basic* hanya menampilkan toolbar standar.



Gambar 2.3 *Toolbar*

2. *Toolbox*

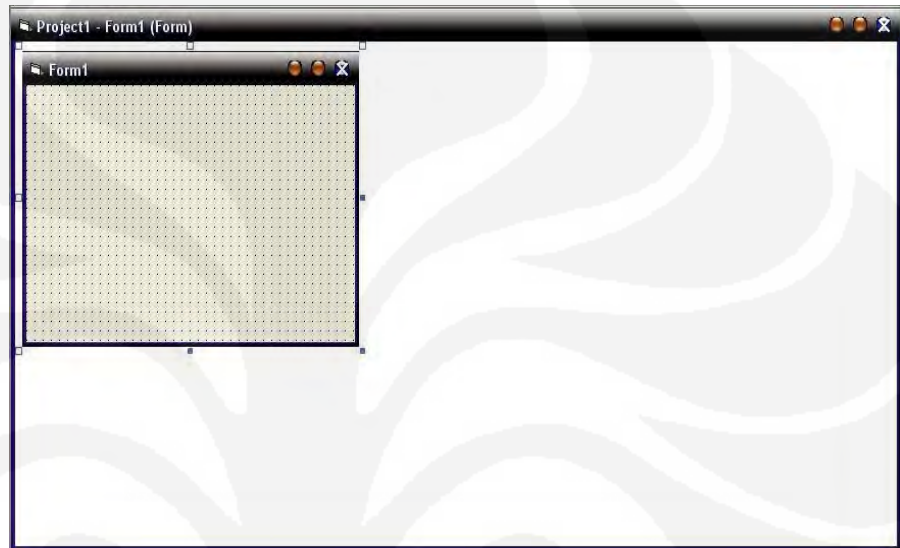
Toolbox adalah suatu objek yang terdiri dari elemen-elemen antarmuka program untuk ditambahkan ke dalam form. *Toolbox* biasanya terletak di sebelah kiri layar dan mengandung kontrol-kontrol yang bisa digunakan untuk menambah gambar, label, tombol, kotak daftar, menu serta bentuk-bentuk geometris pada antarmuka. Setiap kontrol yang ditambahkan kedalam form akan menjadi objek atau elemen antarmuka yang bisa diprogram.



Gambar 2.4 *Toolbox*

3. *Form Window*

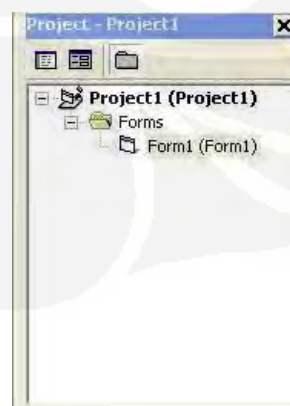
Form Window adalah daerah kerja utama, dimana dapat digunakan untuk membuat program-program aplikasi *Visual Basic*. Pada form ini dapat ditempatkan berbagai macam objek misalnya teks, gambar, tombol-tombol perintah, *database*, *combo box* dan lain-lain.



Gambar 2.5 *Form Window*

4. *Project Explorer*

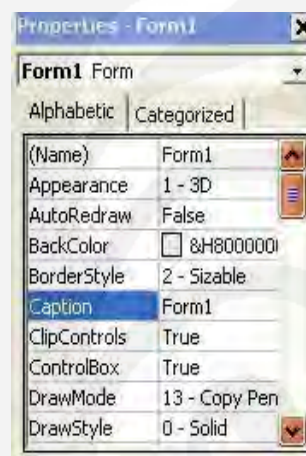
Project Explorer merupakan jendela yang mengandung semua form dan modul di dalam aplikasi *Visual Basic*. Setiap aplikasi dalam *Visual Basic* disebut dengan *project* dan setiap *project* dapat mengandung lebih dari satu form dan modul.



Gambar 2.6 *Project Explorer*

5. Properties Window

Properties window merupakan jendela yang terdiri dari semua informasi mengenai objek yang terdapat pada aplikasi *Visual Basic*. properti adalah sifat sebuah objek, misalnya tampilan, warna, dimensi, huruf dan sebagainya. Jendela properti ini digunakan untuk mengubah karakteristik atau setting properti dari elemen yang terdapat pada form.



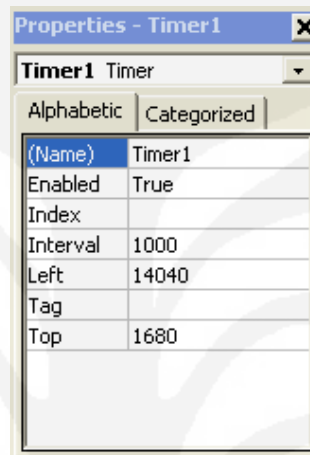
Gambar 2.7 Properties Window

2.2.1 Object Timer

Dalam *Visual Basic* terdapat objek *timer* yang digunakan untuk mengeksekusi sekelompok pernyataan dalam jangka waktu tertentu. Objek *timer* merupakan sebuah *stopwatch* (pengukur waktu) tersembunyi yang memberikan akses ke dalam waktu sistem dari program. Objek *timer* ini juga dapat digunakan untuk menghitung mundur dari waktu tertentu, untuk menyebabkan penundaan pada program atau mengulang suatu aksi pada interval tertentu.

Objek *timer* memiliki keakuratan hingga 1 milidetik atau 1/1000 detik. Walaupun *timer* tidak Nampak ketika program dijalankan, setiap *timer* berhubungan dengan *event procedure* yang berjalan setiap kali waktu interval *timer* berakhir. Interval *timer* diset menggunakan properti interval

dan *timer* diaktifkan dengan mengubah properti *enable* menjadi *True*. Setelah *timer* diaktifkan, maka *timer* akan berjalan terus (menjalankan *event procedure*-nya pada interval yang telah ditentukan) sampai *timer* tersebut dihentikan atau *enable* menjadi *False*. (Micheal Halvorson,2000, chap 2)



Gambar 2.8 *Object Timer*



Gambar 2.9 *Properties Object Timer*

2.2.2 *Object Common Dialog*

Objek *common dialog* berguna untuk menampilkan lima kotak dialog standar pada program. Masing-masing kotak dialog ini bisa ditampilkan dari dalam satu objek *common dialog* menggunakan metode objek *common dialog* yang berhubungan dengan kotak tersebut. Untuk mengendalikan isi dari kotak *common dialog* yaitu dengan mengatur propertinya. (Micheal Halvorson,2000, chap 2). Lima kotak *common dialog* yang disediakan ditampilkan dengan metode-metode yang dapat anda gunakan pada table berikut ini :

Gambar 2.10 *Object Common Dialog*Tabel 2.1 Pilihan pada *Common Dialog*

Kotak Dialog	Tujuan	Metode
<i>Open</i>	Mendapatkan <i>drive</i> , nama folder dan nama file untuk file yang sudah ada.	<i>ShowOpen</i>
<i>Save As</i>	Mendapatkan <i>drive</i> , nama folder dan nama file untuk file baru.	<i>ShowSave</i>
<i>Print</i>	Memberikan pilihan pencetakan untuk pemakai	<i>ShowPrinter</i>
<i>Font</i>	Memberikan pilihan jenis dan gaya <i>font</i>	<i>ShowFont</i>
<i>Color</i>	Memberikan pilihan jenis-jenis warna dari palet.	<i>ShowColor</i>

Sumber : (Micheal Halvorson, *Microsoft Visual Basic 6.0 Professional Step by Step*, 2000, chap 2)

2.2.3 Tipe Data

Pada umumnya, tipe data *variant* adalah satu-satunya tipe data yang dibutuhkan. Variabel *variant* dapat menyimpan semua tipe data dasar *Visual Basic* (yang sudah didefinisikan) dan dapat berpindah-pindah format secara otomatis. Variabel juga mudah digunakan dan tidak perlu memikirkan ukuran variable apabila dideklarasikan. Namun jika ingin membuat yang

sangat cepat dan teliti, maka perlu menggunakan jenis-tipe data yang sesuai dengan kebutuhan. (Micheal Halvorson,2000, chap 3)

Tabel 2.2 Tipe data

Tipe Data	Ukuran	Kisaran
<i>Integer</i>	2 byte	-32,768 sampai 32,767
<i>Long Integer</i>	4 byte	-2,147,483,648 sampai 2,147,483,647
<i>Single-precision floating point</i>	4 byte	-3.402823E38 sampai 3.402823E38
<i>Double-precision floating point</i>	8 byte	-1.79769313486232E308 sampai 1.79769313486232E308
<i>Currency</i>	8 byte	-922337203685477.5808 sampai 922337203685477.5808
<i>String</i>	1 byte per karakter	0 sampai 65,535
<i>Boolean</i>	2 byte	<i>True</i> atau <i>False</i> (1 atau 0)
<i>Date</i>	8 byte	1 Januari 100 sampai 31 Desember 9999
<i>Variant</i>	16 byte (untuk angka); 22 byte + 1 byte per karakter (untuk <i>string</i>)	Semua tipe data

Sumber : (Micheal Halvorson, *Microsoft Visual Basic 6.0 Professional Step by Step*, 2000, chap 3)

2.2.4 Struktur Keputusan *IF....Then*

Struktur keputusan *If Then* digunakan untuk menyeleksi kondisi. Struktur keputusan ini berfungsi mengevaluasi sebuah kondisi pada program dan melakukan tindakan berdasarkan hasilnya. Dalam bentuk yang sederhana, struktur keputusan *If....Then* ditulis dalam satu baris tunggal :

If condition Then statement

Dimana *condition* adalah ekspresi berkondisi dan *statement* adalah pernyataan program *Visual Basic* yang valid. (Micheal Halvorson,2000, chap 6)

2.2.5 Loop For Next

Loop For.... Next berfungsi untuk mengeksekusi sekelompok pernyataan program selama beberapa kali dalam suatu *event* prosedur. Pendekatan ini sangat bermanfaat apabila ingin melakukan beberapa perhitungan yang berhubungan, bekerja dengan elemen pada layar, atau memproses beberapa input dari pemakai. *Loop For* dan *Next* sebenarnya hanyalah merupakan penulisan singkat dari pernyataan program yang panjang. Karena setiap kelompok pernyataan program tersebut pada dasarnya melakukan hal yang sama.

Contoh sintaks untuk *loop For* dan *Next* adalah sebagai berikut :

For variable = start To end

Statements to be repeated

Next variable

(Micheal Harverson,2000, chap 8)

2.3 Microsoft Access 2007

Microsoft Access adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi *Microsoft Office*, selain tentunya *Microsoft Word*, *Microsoft Excel* dan *Microsoft PowerPoint*. Aplikasi ini menggunakan mesin basis data *Microsoft Jet Database Engine* dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna.

Microsoft Access dapat menggunakan data yang disimpan di dalam format *Microsoft Access*, *Microsoft Jet Database Engine*, *Microsoft SQL Server*, *Oracle Database* atau semua container basis data yang mendukung standar ODBC (*Open Database Connectivity*). Para programmer yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. *Microsoft Access* juga mendukung teknik-teknik pemrograman berorientasi objek tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

Microsoft Access digunakan kebanyakan oleh bisnis-bisnis kecil dan menengah, di dalam sebuah organisasi yang kecil bahkan mungkin juga digunakan oleh perusahaan yang cukup besar, dan juga para programmer untuk membuat sebuah sistem buatan sendiri untuk menangani pembuatan dan manipulasi data. *Access* juga dapat digunakan sebagai sebuah basis data untuk aplikasi *Web* dasar yang disimpan di dalam *server* yang menjalankan *Microsoft Internet Information Services* (IIS) dan menggunakan *Microsoft Active Server Pages* (ASP).

Microsoft Access kurang begitu bagus jika diakses melalui jaringan sehingga aplikasi-aplikasi yang digunakan oleh banyak pengguna cenderung menggunakan solusi sistem manajemen basis data yang bersifat klien/server. Meskipun demikian, tampilan muka *Access* (*form*, *report*, *query*, dan kode *Visual Basic*) yang dimilikinya dapat digunakan untuk menangani basis data yang sebenarnya diproses oleh sistem manajemen basis data lainnya, seperti halnya *Microsoft Jet Database Engine* (yang secara *default* digunakan oleh *Microsoft Access*), *Microsoft SQL Server*, *Oracle Database*, dan beberapa produk lainnya yang mendukung ODBC. (Sumber : wikipedia.org)

2.4 PLC

Programmable Logic Controller (PLC) pada dasarnya adalah sebuah komputer yang khusus dirancang untuk mengontrol suatu proses atau mesin. Proses yang dikontrol ini dapat berupa regulasi variabel secara kontinyu seperti pada sistem-sistem servo atau hanya melibatkan kontrol dua keadaan (On/Off) saja tapi dilakukan secara berulang-ulang seperti umum kita jumpai pada mesin pengeboran, sistem konveyor, dan lain sebagainya. (Iwan Setiawan, 2006, chap 1, p.187)

Penggunaan PLC di bidang perindustrian membuat PLC memiliki beberapa karakteristik, antara lain :

1. Kokoh dan dirancang untuk tahan terhadap getaran, suhu, kelembaban dan kebisingan.
2. *Interface* untuk *input* dan *output* telah tersedia secara built-in di dalamnya.
3. Mudah diprogram dan menggunakan bahasa pemrograman yang mudah dipahami dan sebagian besar berkaitan dengan operasi-operasi logika dan penyambungan (*switching*).
4. Dapat menangani *input* dan *output* dalam jumlah besar dan dalam bentuk sinyal analog maupun digital.

Pada umumnya, sebuah sistem PLC memiliki lima komponen dasar, yaitu:

1. Unit processor atau *Central Processing Unit (CPU)* adalah unit yang berisi mikroprosesor yang menginterpretasikan sinyal-sinyal input dan melaksanakan pengontrolan, sesuai dengan program yang disimpan di dalam memori, lalu mengkomunikasikan keputusan-keputusan yang diambilnya sebagai sinyal-sinyal kontrol ke *interface output*.
2. Unit catu daya diperlukan untuk mengkonversi tegangan AC sumber menjadi tegangan rendah DC (5V dan 24 V) yang dibutuhkan oleh prosesor dan rangkaian-rangkaian di dalam modul-modul antarmuka input dan output.

3. Perangkat pemrograman dipergunakan untuk memasukan program yang dibutuhkan ke dalam memori. PLC sekarang kebanyakan sudah menggunakan program melalui software untuk memasukan program yang dibuat ke dalam PLC.
4. Unit memori adalah tempat program yang digunakan untuk melaksanakan tindakan-tindakan pengontrolan oleh mikroprosesor disimpan.
5. Bagian *input* dan *output* adalah antarmuka dimana prosesor menerima informasi dari dan mengkomunikasikan informasi kontrol ke perangkat-perangkat eksternal. Sinyal-sinyal input dapat berasal dari saklar-saklar serta sensor-sensor lain seperti misalnya sel-sel *fotoelektrik*, sensor suhu atau sensor aliaran cairan, dsb. Sinyal-sinyal *output* mungkin diberikan pada kumparan-kumparan starter motor, katup-katup, *solenoid*, dll. Ada dua tipe I/O pada PLC yaitu I/O digital dan I/O analog. Pada I/O *digital*, *input* dan *output*-nya dapat menerima dan menghasilkan sinyal *digital* yang berbentuk biner „1’ dan „0’ atau kondisi *on-off*, sebagai contoh saklar *on-off* sebagai *input digital* dan *output*-nya mengontrol *on-off solenoid*. Sedangkan, pada I/O analog, *input* atau *output* analognya dapat menerima dan menghasilkan sinyal analog yang tidak hanya *on-off* saja.

2.4.1 PLC Omron CJ1M-CPU11

CJ1M-CPU11 merupakan bagian yang berfungsi sebagai *Control Processing Unit* dari PLC Omron seri CJ. CJ1M-CPU11 memiliki kapasitas memori 5 Ksteps dan mempunyai hingga 160 I/O point. CJ1M-CPU11 ini digunakan untuk aplikasi sederhana dan tidak begitu kompleks karena keterbatasan memori. Untuk komunikasi CJ1M-CPU11 ini dengan komputer digunakan komunikasi serial RS-232. Kemudian untuk pemrograman *ladder* diagramnya digunakan *software CX-Programmer*. (Sumber : PLC Omron CJ1M CPU11 datasheet)



Gambar 2.11 CJ1M-CPU11

2.4.2 PLC Input CJ1W-ID211

CJ1W-ID211 merupakan bagian yang berfungsi sebagai modul *input* PLC. CJ1W-ID211 ini merupakan modul *digital input* karena modul ini memiliki spesifikasi untuk sinyal-sinyal digital dan tidak dapat digunakan untuk sinyal analog. CJ1W-ID211 memiliki I/O point 16 *input*, dan memiliki *input voltage* dan *current* 24 VDC dan 7 mA. (Sumber : PLC Omron *input* CJ1W-ID211 datasheet)



Gambar 2.12 PLC Input CJ1W-ID211

2.4.3 PLC Output CJ1W-OD211

CJ1W-OD211 merupakan bagian yang berfungsi sebagai modul *output* PLC. CJ1W-OD211 hanya bekerja menerima hasil keluaran instruksi dari CPU unit berupa ON/OFF untuk mengontrol *device* eksternal. CJ1W-OD211 terdiri dari 16 *output point* serta memiliki *Rated Voltage* 12 – 24 VDC dan memiliki *Maximum Load Current* 0.5 A/point dan 5 A/unit. (Sumber : PLC Omron *output* CJ1W-OD211 datasheet)



Gambar 2.13 PLC Output CJ1W-OD211

2.4.4 PLC Power Supply CJ1W-PA202

CJ1W-PA202 merupakan bagian yang berfungsi sebagai *power supply* untuk unit CPU. CJ1W-PA202 memiliki spesifikasi *supply* tegangan 110 – 240 VAC dan memiliki total konsumsi daya kurang lebih 14 W. .(Sumber : PLC Omron *power supply* CJ1W-PA202 datasheet)



Gambar 2.14 PLC Power Supply CJ1W-PA202

2.5 Komunikasi Microsoft Visual Basic 6.0 dengan PLC

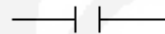
PLC dapat berkomunikasi dengan komputer dengan menggunakan *Host Link*. *Host Link* ini merupakan *interface* PLC terhadap *host* komputer. Melalui *Host Link* seluruh area memori dalam PLC dapat diakses termasuk memori program. Dalam tugas akhir ini digunakan *Host link* RS-232 yang dapat dihubungkan dengan *host* komputer. *Software* yang digunakan untuk menghubungkan dan membuka port PLC Omron CPU 11 dengan komputer

adalah Omron *FinsGateway* sedangkan untuk mengenali komponen-komponen yang ada di dalam PLC Omron CPU11 berupa I/O dan data memori, digunakan SYSMAC *Compolet*. Software ini digunakan untuk mempermudah PLC berkomunikasi dengan komputer.

2.6 CX-Programmer 7

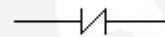
Program ini digunakan untuk membuat *ladder diagram* pada PLC Omron CJ1M. dibawah ini terdapat beberapa instruksi yang ada pada CX-Programmer 7 :

1. Kontak NO (*Normaly Open*)



Gambar 2.15 Kontak NO

2. Kontak NC (*Normaly Close*)



Gambar 2.16 Kontak NC

3. *Coil*

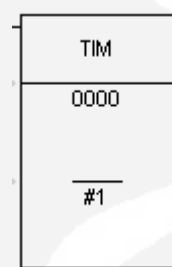
Digunakan seabail *coil relay* bantu dan juga *output*.



Gambar 2.17 *Coil*

4. *Timer*

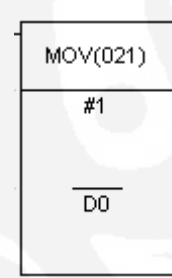
Digunakan untuk mengatur waktu *On-Off* dari kontak *timer* tersebut, sesuai dengan waktu *setting* dari *timer* tersebut. #1 pada gambar dimaksud waktu *setting timer* adalah 0.1 s



Gambar 2.18 *Timer*

5. *MOV*

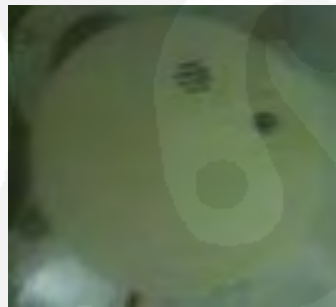
Instruksi ini digunakan untuk memindahkan angka *decimal*, biner dan lain-lain ke alamat tertentu. Pada gambar dibawah dimaksudkan memindahkan angka 1 ke dalam data memori 0 (D0).



Gambar 2.19 *MOV*

2.7 Detektor Asap (*Smoke Detector*)

Detektor asap adalah alat yang berfungsi untuk mendeteksi asap, prinsip kerja detektor asap ini berdasarkan sensor ionisasi dan fotoelektrik. Pada sensor ionisasi berisi sejumlah kecil bahan amerisium yang dilekatkan pada suatu lembar matriks emas di dalam suatu kamar ionisasi. Amerisium pada detector asap akan mengionisasikan udara di dalam kamar (*chamber*) pengindera, memberikan daya konduksi dan suatu aliran arus melalui udara antara dua muatan elektroda. Hal ini memberikan kamar pengindera suatu efek aliran listrik. Apabila partikel asap masuk daerah ionisasi, maka asap tersebut akan mengurangi aliran listrik udara dengan menempelkan diri pada ion yang menyebabkan pengurangan gerak ion. Ketika arus listrik kurang dari tingkat yang ditetapkan, maka detektor akan merespon dan membunyikan *alarm*. (Sumber : wikipedia.org)



Gambar 2.20 *Smoke Detector*

2.8 Sensor Gerak (*Motion Sensor*)

Sensor gerak merupakan sensor yang mendeteksi gerakan benda atau makhluk hidup. Salah satunya sensor gerak yaitu *passive Infrared* digunakan kebanyakan pada lampu dan sistem keamanan adalah sebuah sistem pasif yang dapat mendeteksi energi inframerah. Oleh karena itu sensor ini dikenal sebagai PIR (*Passive InfraRed*) *pyroelectric detektor* atau sensor. Sering kali, teknologi PIR akan dipasangkan dengan model lain untuk memaksimalkan

akurasi dan mengurangi penggunaan energi. Untuk membuat sebuah sensor yang dapat mendeteksi manusia, perlu dibuat sensor yang sensitif terhadap suhu tubuh manusia. Manusia, memiliki suhu kulit sekitar 93 derajat F, memancarkan energi inframerah dengan panjang gelombang antara 9 dan 10 mikrometer. Oleh karena itu, biasanya sensor sensitif pada kisaran 8-12 mikrometer. (Sumber : wikipedia.org)



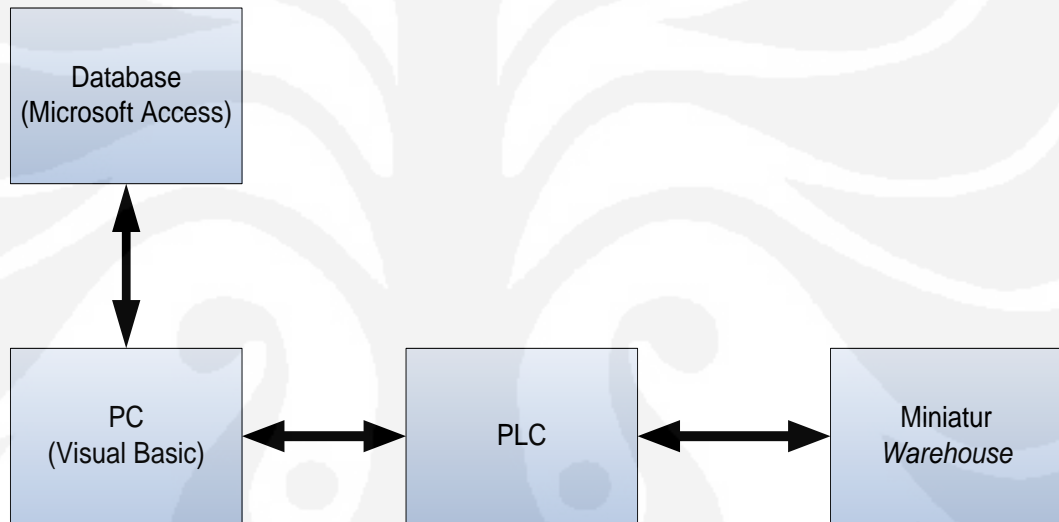
Gambar 2.21 Sensor Gerak *Passive InfraRed*

BAB 3

PERANCANGAN SISTEM

3.1 Cara Kerja Sistem

Secara keseluruhan cara kerja dari sistem ini dapat digambarkan melalui blok diagram dibawah ini :



Gambar 3.1 Blok diagram cara kerja sistem

Sistem ini merupakan aplikasi dari SCADA sederhana yang menggunakan satu PLC untuk mengawasi dan mengendalikan miniatur *warehouse*. Dalam diagram tersebut digambarkan PLC dapat dikontrol dengan komputer dengan menggunakan HMI SCADA, HMI SCADA ini dibuat dengan bahasa pemrograman *Visual Basic 6.0*. Kemudian untuk penyimpanan *database*-nya digunakan *Microsoft Access 2007*.

Kemudian beban yang akan dikontrol dan diawasi yaitu lampu dan *fan* yang terdapat dalam miniatur *warehouse*. Dalam miniatur *warehouse* ini terdapat 4 buah lampu sebagai unit penerangan miniatur *warehouse* dan 2 buah *fan* sebagai unit pendingin ruangan. Lalu dalam miniatur *warehouse* ini juga terdapat sensor pendeteksi asap (*Smoke Detector*) dan sensor gerak (*Motion Sensor*). Kedua sensor ini juga akan dikontrol oleh sistem SCADA melalui HMI SCADA.

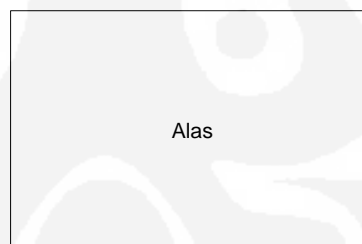
3.2 Perancangan Miniatur *Warehouse*

Pada perancangan ini akan dibuat sebuah miniatur *warehouse* dengan menggunakan bahan *acrylic*. Disini digunakan bahan *acrylic* karena lebih praktis dalam proses pembuatannya.

Desain miniatur *warehouse* :

- 1) Alas;

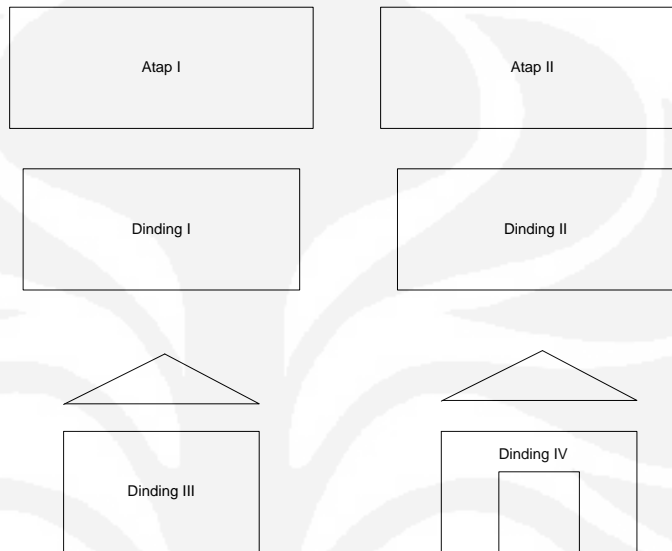
Alas miniatur ini di desain seperti berikut dengan ketebalan 3 mm.



Gambar 3.2 Desain alas miniatur *warehouse*

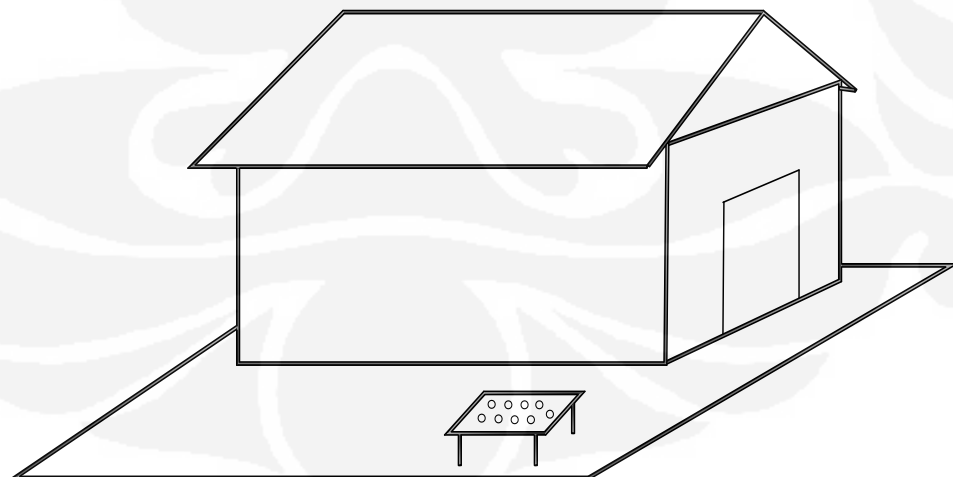
2) Bangunan;

Bangunan miniatur *warehouse* terdiri dari beberapa dinding dan atap dengan ketebalan 3 mm.



Gambar 3.3 Desain bangunan miniatur *warehouse*

3) Desain secara keseluruhan;



Gambar 3.4 Desain keseluruhan miniatur *warehouse*

Pada miniatur *warehouse* ini terdapat beberapa unit beban dan sensor yaitu :

a) Unit Penerangan (*Lighting*)

Pada bagian ini digunakan 4 buah LED sebagai lampu penerangan dalam miniatur *warehouse*.

b) Unit Pendingin (*Cooling*)

Pada bagian ini digunakan 2 buah *fan* dengan tegangan kerja 12 VDC.

c) Sensor Pendeteksi Asap (*Smoke Detector*)

Pada bagian ini digunakan *Smoke Detector Kid* dengan tegangan kerja 12 VDC. Sensor ini digunakan untuk mendeteksi asap penyebab kebakaran didalam miniatur *warehouse*.

d) Sensor gerak (*Motion Sensor*)

Pada bagian ini digunakan sensor gerak model *Heles*, dengan tegangan kerja 6 – 12 VDC. Sensor ini digunakan sebagai sistem keamanan untuk pendeteksi gerak apabila ada pencurian di malam hari.

3.3 Perancangan *Software*

Perancangan software ini terdiri dari dua bagian yaitu perancangan *ladder diagram* untuk menjalankan PLC dan perancangan HMI SCADA. Perancangan *software* ini saling terkait satu sama lain, karena tanpa perancangan *ladder diagram* terlebih dahulu nanti akan mengalami kesulitan dalam membuat *software* HMI SCADA-nya. Perancangan *ladder diagram* ini menggunakan *software CX-Programmer* versi 7 keluaran Omron. Kemudian untuk membuat *software* HMI SCADA-nya disini digunakan *Microsoft Visual Basic* 6.0. Bahasa pemrograman *Visual Basic* merupakan bahasa pemrograman yang berorientasi objek, oleh karena itu cukup mudah dan praktis untuk digunakan dalam mendesain HMI SCADA.

3.3.1 Perancangan *Ladder Diagram*

Dalam perancangan *ladder diagram* ini terlebih dahulu ditentukan kebutuhan I/O PLC yang dipakai. Penentuan nomor I/O ini penting dilakukan untuk memudahkan dalam pembuatan *ladder diagram*, agar tidak terjadi kesalahan dalam pemrograman. Untuk gambar *ladder diagram* secara lengkap terdapat pada lampiran. Berikut ini adalah I/O PLC yang dipakai.

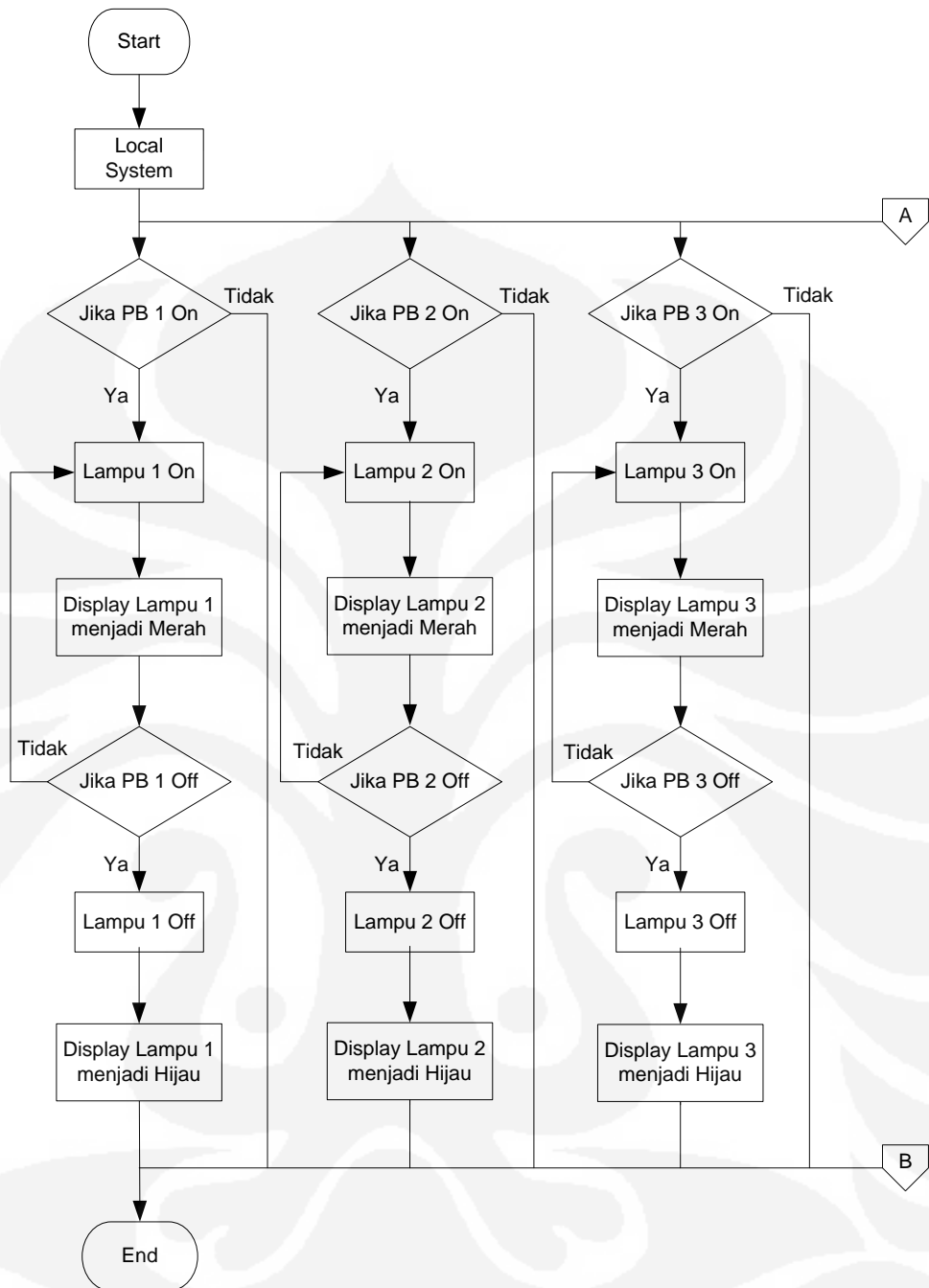
Tabel 3.1 *Input* PLC CJ1W-ID211 yang digunakan

No	Input	Keterangan
1	0.00	<i>Push Button</i> Lampu 1
2	0.01	<i>Push Button</i> Lampu 2
3	0.02	<i>Push Button</i> Lampu 3
4	0.03	<i>Push Button</i> Lampu 4
5	0.04	<i>Push Button</i> Fan 1
6	0.05	<i>Push Button</i> Fan 2
7	0.06	<i>Push Button</i> Reset Alarm
8	0.07	<i>Push Button</i> Reset System
9	0.08	<i>Push Button</i> Emergency
10	0.09	Relay Smoke Detector
11	0.10	Relay Motion Sensor

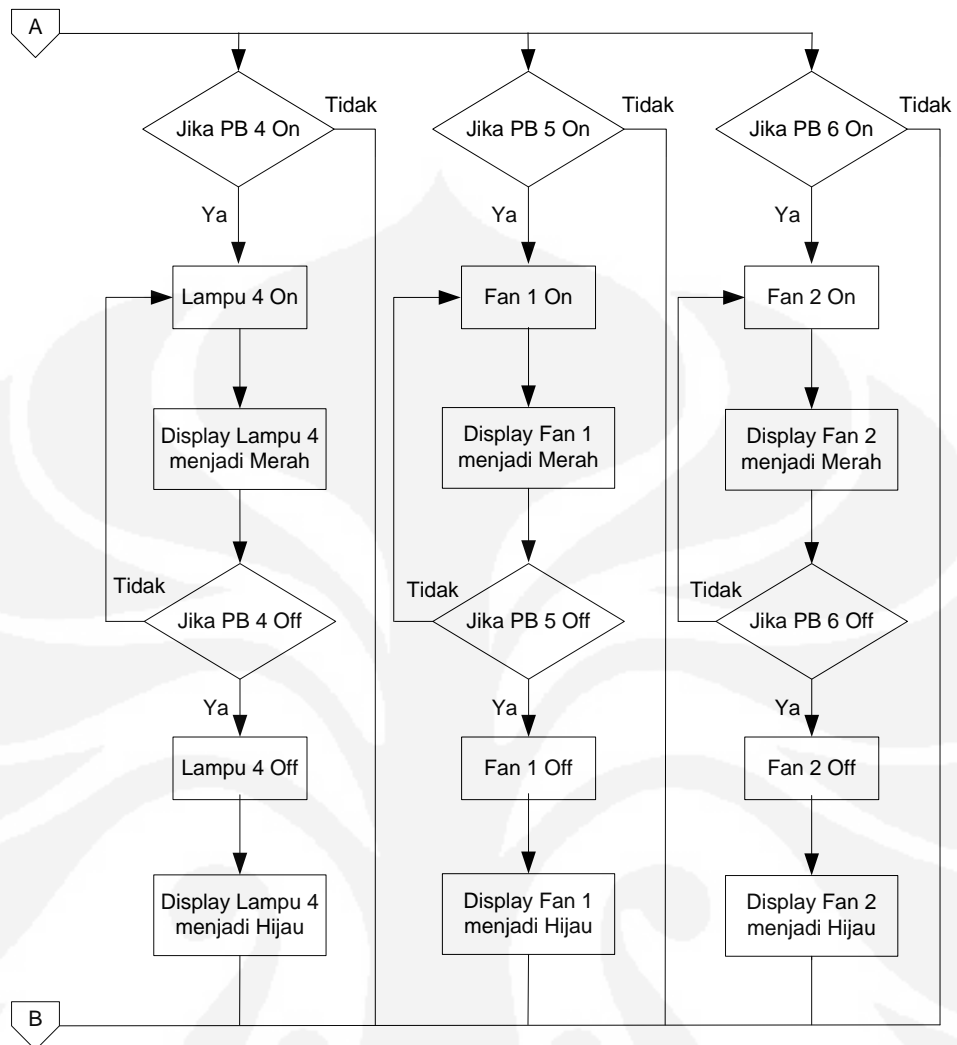
Tabel 3.2 *Output* PLC CJ1W-OD211 yang digunakan

No	Output	Keterangan
1	1.00	Lampu 1
2	1.01	Lampu 2
3	1.02	Lampu 3
4	1.03	Lampu 4
5	1.04	<i>Fan 1</i>
6	1.05	<i>Fan 2</i>
7	1.06	<i>Alarm</i>
8	1.07	<i>Relay ON/OFF Smoke Detector</i>
9	1.08	<i>Relay ON/OFF Motion Sensor</i>

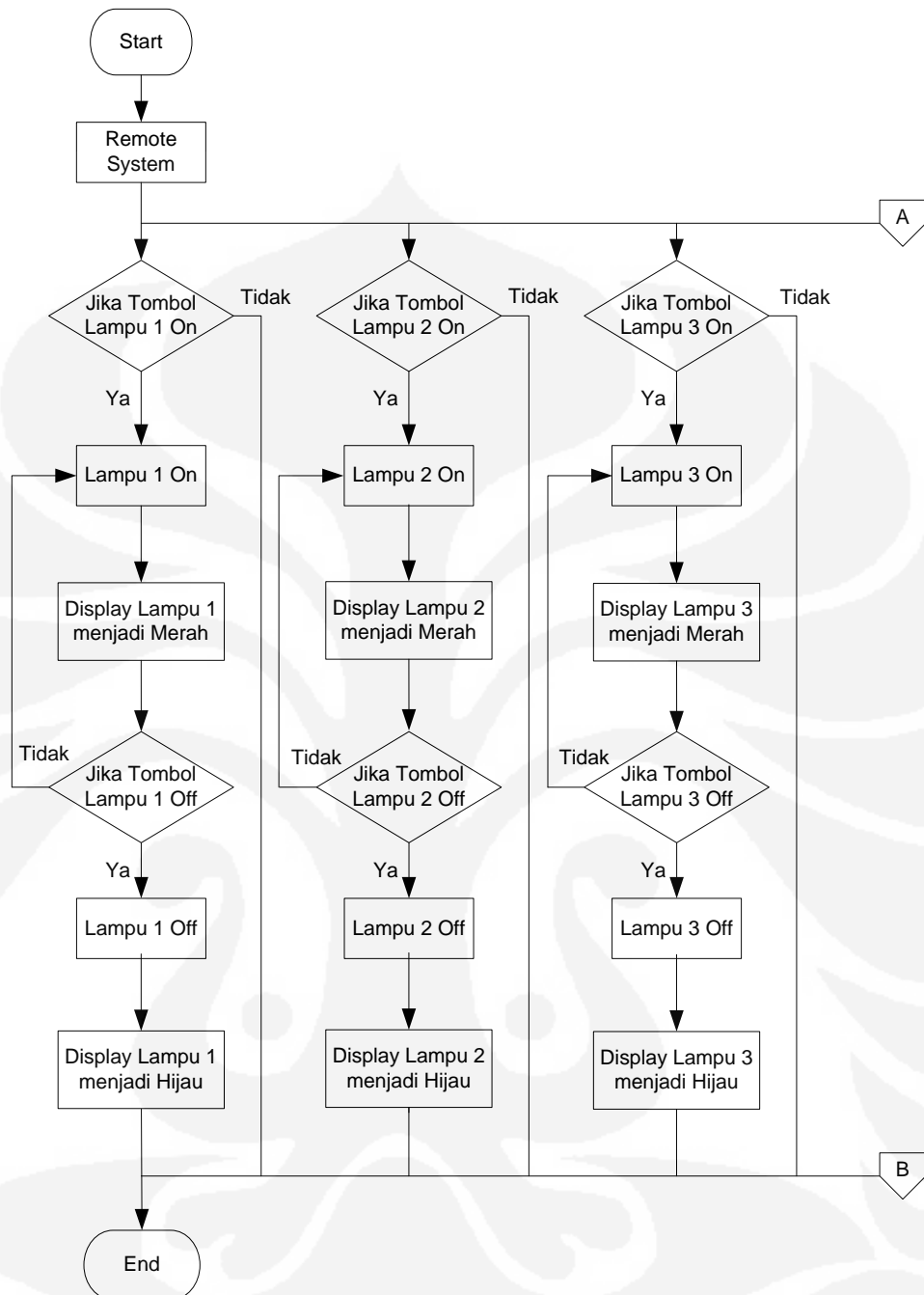
Setelah menentukan *input* dan *output* PLC, selanjutnya membuat *flowchart* agar memudahkan dalam pembuatan *ladder diagram*-nya. *Flowchart* ini merupakan algoritma cara kerja sistem. Dalam sistem yang akan dirancang nanti terdapat dua *mode* yaitu *local* dan *remote*. Berikut ini adalah *flowchart* cara kerja kedua sistem tersebut.



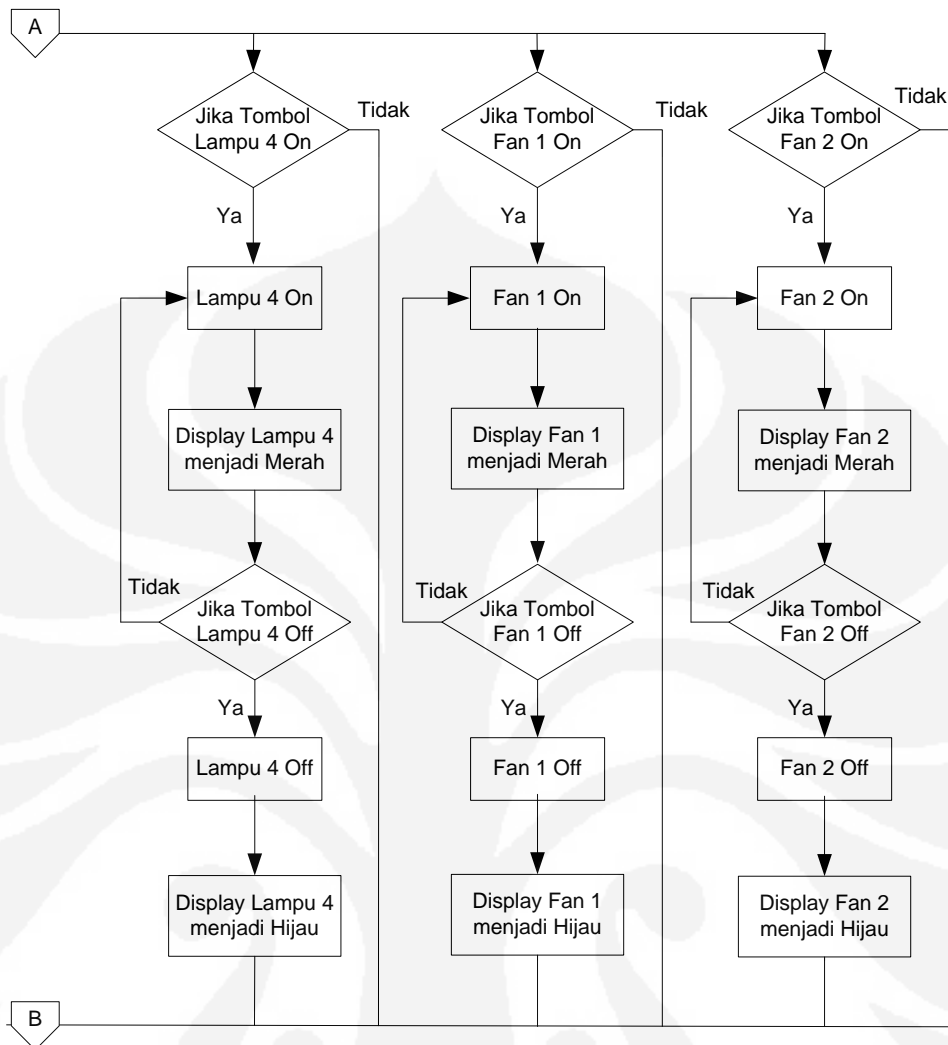
Gambar 3.5 Flowchart Local System



Gambar 3.6 *Flowchart Local System (Lanjutan)*



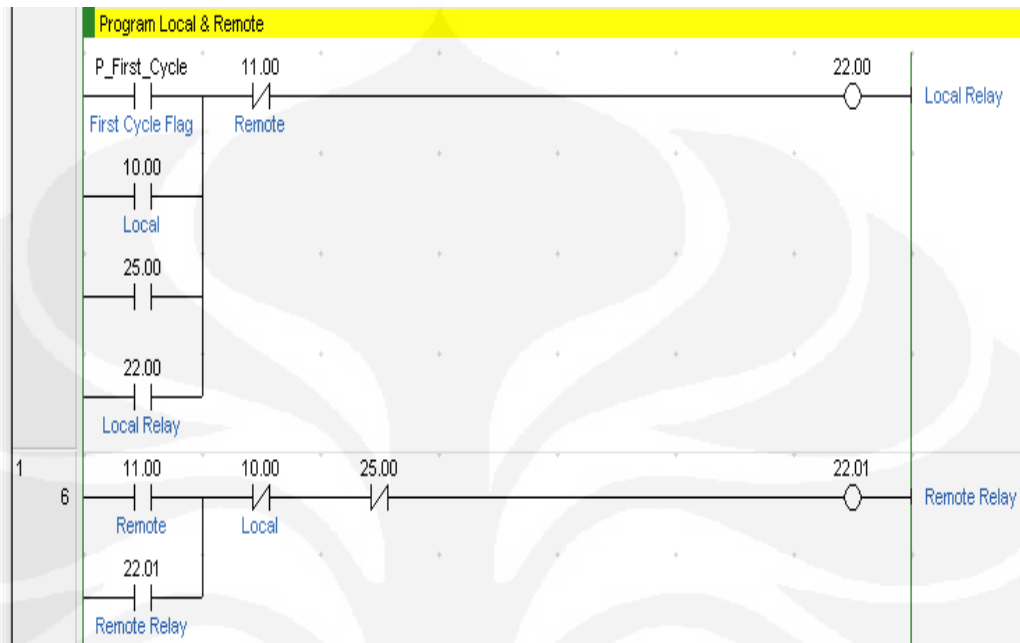
Gambar 3.7 Flowchart Remote System



Gambar 3.8 *Flowchart Remote System (Lanjutan)*

Flowchart local system dan *remote system* ini merupakan diagram alir dari cara kerja mode *local* dan *remote* yang terdapat pada HMI SCADA, dalam pengoperasiannya kedua mode ini pada dasarnya sama yaitu pengontrolan lampu dan *fan* pada miniatur *warehouse*. Akan tetapi untuk mode *local* pengontrolan bebannya melalui *push button* yang ada di miniatur *warehouse*, sedangkan pada mode *remote* pengontrolan bebannya melalui HMI SCADA yang ada di komputer.

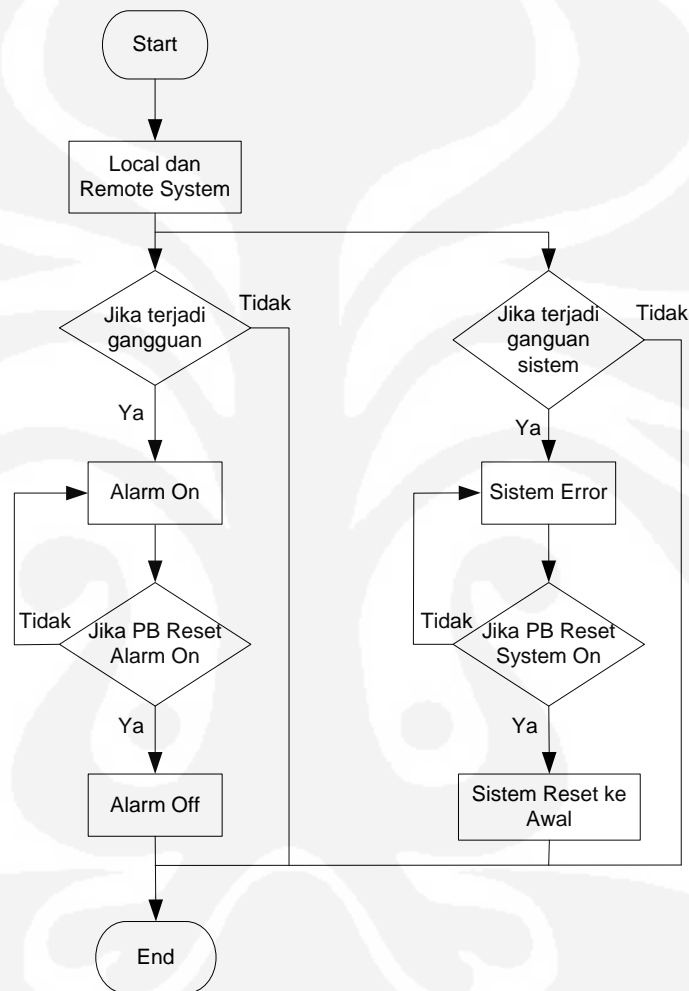
Berikut ini gambar *ladder diagram* dari fungsi mode *local* dan *remote* :



Gambar 3.9 *Ladder diagram* mode *local* dan *remote*

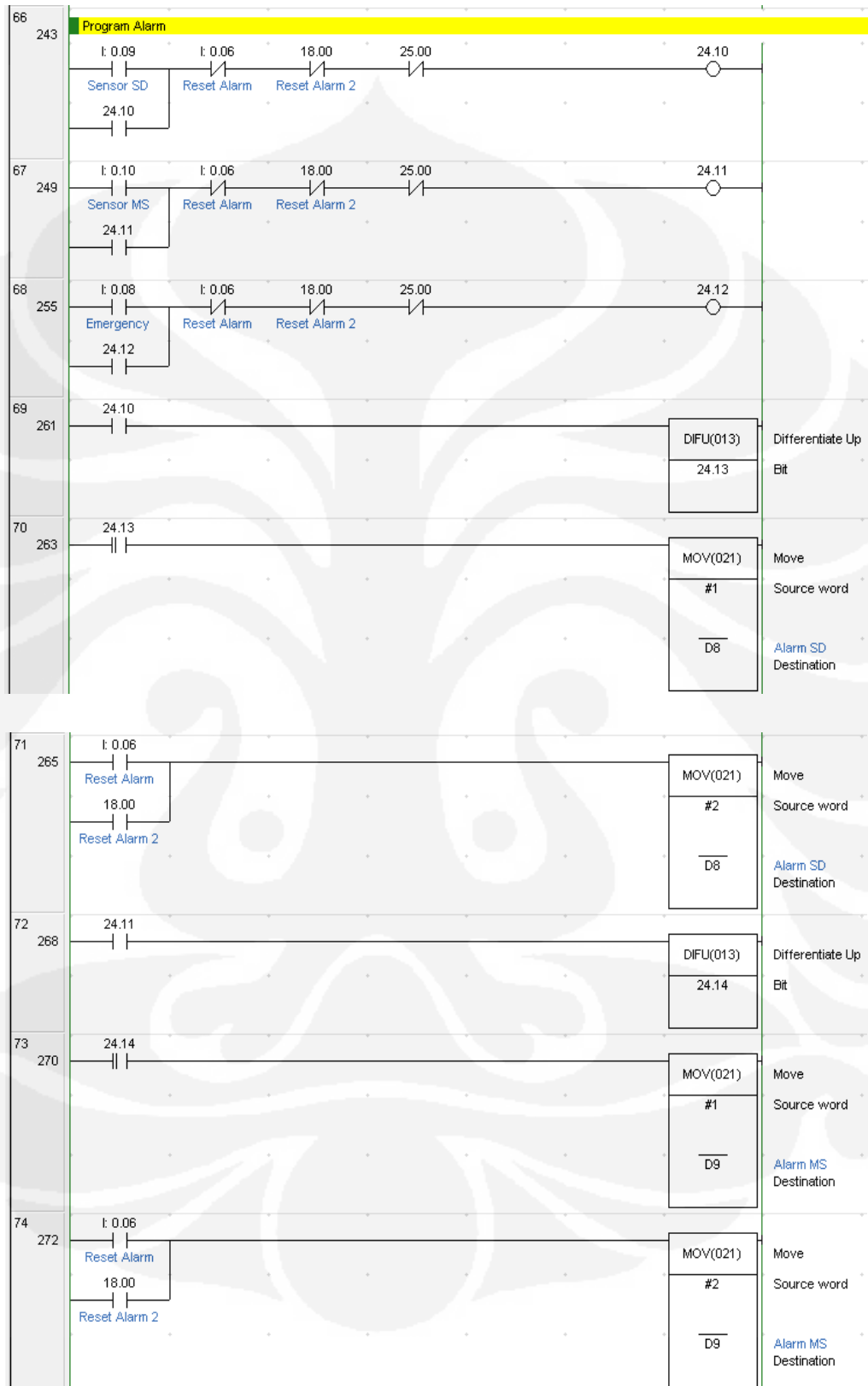
Pada gambar *ladder diagram* diatas terdapat dua *coil* yaitu 22.00 sebagai *local relay* atau *relay* bantu untuk mode *local* dan *coil* 22.01 sebagai *remote relay* atau *relay* bantu untuk mode *remote*. Kontak 25.00 merupakan kontak *reset system*. Lalu pada perancangan *ladder diagram* ini kondisi awal dibuat pada mode *local*, hal ini disebabkan kontak P_First_Cycle yaitu kontak ini akan bekerja ketika PLC menyala dan pada mode *Run Program*.

Kemudian pada perancangan ini terdapat juga sistem keamanan dan pendeteksi kebakaran, karena pada miniatur *warehouse* tersebut terdapat sensor *smoke detector* dan *motion sensor*. Berikut ini *flowchart* untuk cara kerja sistem tersebut.

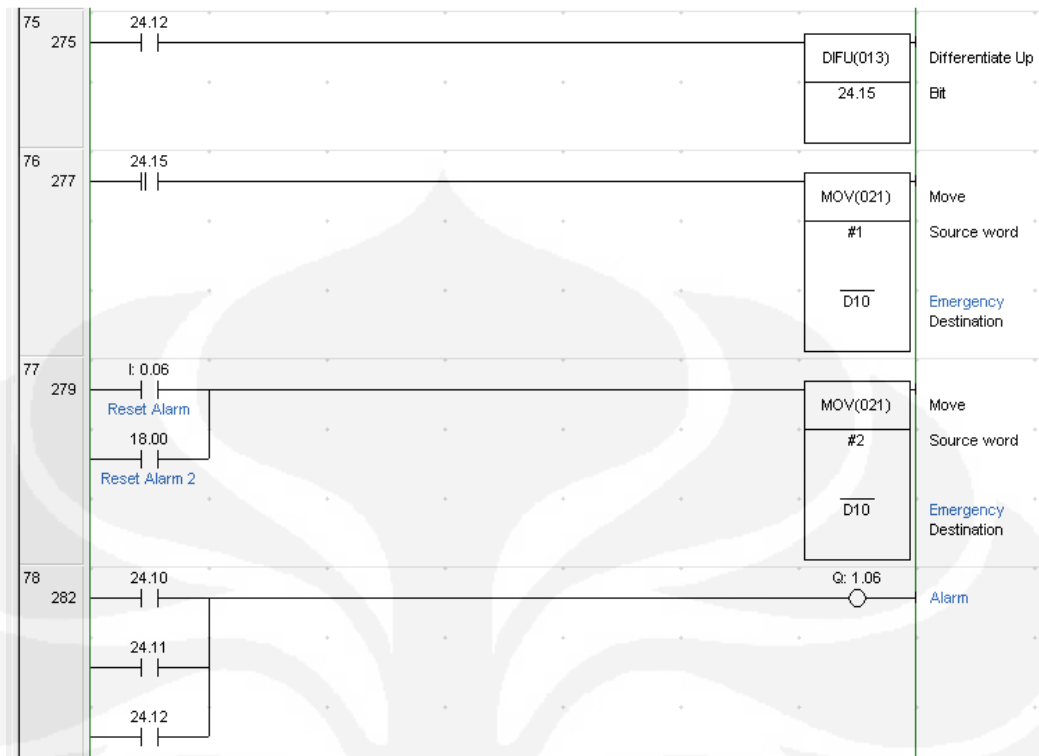


Gambar 3.10 *Flowchart* untuk sistem gangguan

Berikut ini gambar *ladder diagram* untuk program *alarm* :



Gambar 3.11 *Ladder diagram* program *alarm*



Gambar 3.12 *Ladder diagram* program alarm (Lanjutan)

Pada *ladder diagram* diatas dapat dijelaskan sebagai berikut, I0.09 digunakan untuk masukan dari *smoke detector* (SD) dan I0.10 digunakan untuk masukan dari *motion sensor* (MS), lalu I0.08 digunakan untuk masukan dari *push button emergency*. Kemudian DIFU 24.13 dimaksudkan untuk memberikan sinyal bit 1 kepada MOV #1 D8, instruksi ini maksudnya yaitu apabila dapat sinyal bit 1 maka angka 1 dipindahkan ke data memori 8 (DM 8) pada PLC hal ini digunakan untuk menandakan *smoke detector* sedang bekerja. Kemudian I0.06 dan 18.00 digunakan untuk *reset alarm* atau untuk mematikan relai *alarm* untuk kedua sensor, lalu digunakan untuk memberi sinyal bit 1 ke MOV #2 D8, MOV #2 D9 dan MOV #2 D10, hal ini dimaksudkan untuk memindahkan angka 2 ke D8,D9 dan D10 yang digunakan untuk menandakan *alarm* OFF.

3.3.2 Perancangan *Software* HMI

Human Machine Interface (HMI) merupakan bagian yang ada dalam sistem SCADA, HMI menampilkan gambar animasi miniatur *warehouse* yang terdiri dari unit-unit beban yang akan diawasi dan dikontrol. HMI ini digunakan untuk mengawasi dan mengontrol miniatur *warehouse* dari jarak jauh dengan menggunakan komputer. Untuk *source code* program pada masing-masing *form* bisa dilihat di lampiran.

3.3.2.1 *Form Loading*

Form loading ini digunakan sebelum masuk ke *form* utama atau *main program*.

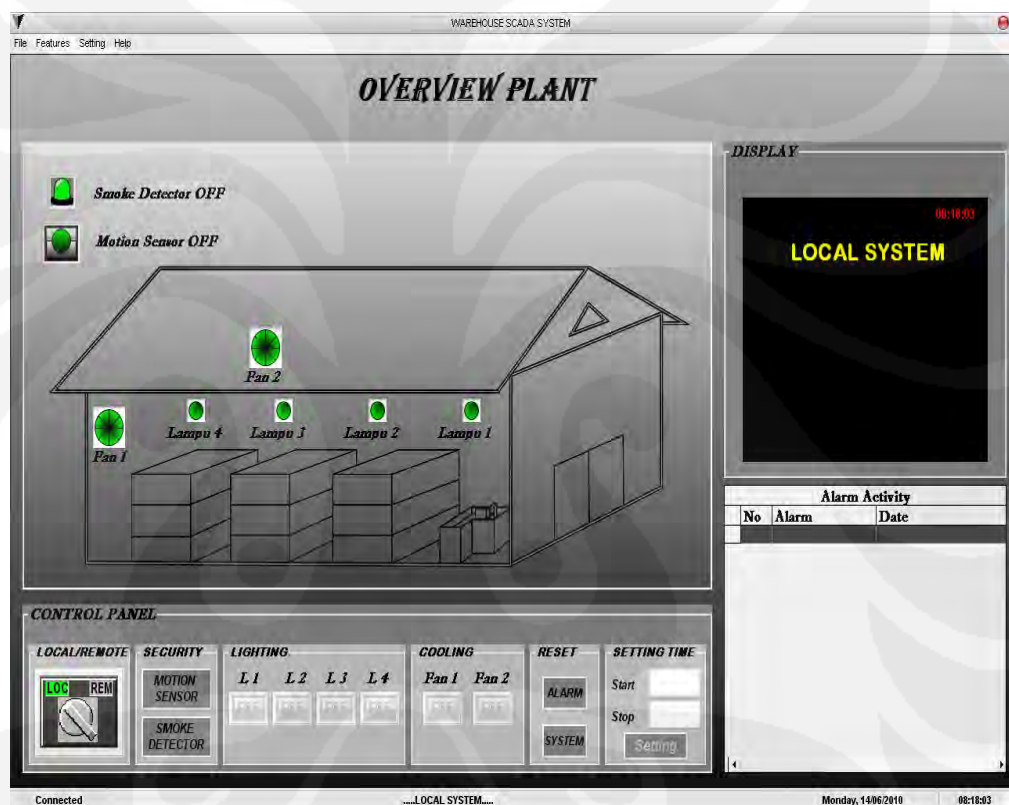


Gambar 3.13 *Form Loading*

Pada *form* ini digunakan dua buah *timer*. *Timer* pertama digunakan untuk melakukan penambahan pada *progressBar* dan *timer* kedua digunakan untuk animasi pada label *loading*.

3.3.2.2 Form Main Program

Pada *form main program* ini akan ditampilkan simulasi dari miniatur *warehouse*. Pada *form* ini terdapat bagian untuk *monitoring* dan *controlling* unit-unit beban yang ada pada miniatur *warehouse*. Selain itu dalam *form* ini terdapat juga *form* lain seperti *form time usage*, *setting*, *history graph* dan *help*. Pada *form* ini ditampilkan *overview* dari miniatur *warehouse*, *alarm activity* dan tombol-tombol kontrol.



Gambar 3.14 *Form Main Program*

3.3.2.3 Form Time Usage

Pada *form time usage* ini akan ditampilkan waktu pemakaian dari masing-masing unit beban yang dikontrol. Dalam *form* ini terdapat beberapa *button* yaitu *Save All Data*, *Delete All Data* dan *Close*. Kemudian untuk masing-masing *hours* meter terdapat *button Save*.

WAREHOUSE SCADA SYSTEM
TIME USAGE

Notice : Save Data When All Unit Turn Off
: Data Automatic Saved On 00:00 AM (OFF)

LAMPU 1	No	Hours	Date
1	0.0133	01/06/2010	
2	0.235	05/06/2010	
3	0.0239	05/06/2010	
4	0.2883	09/06/2010	
5	0.4464	10/06/2010	
6	0.4464	10/06/2010	
7	0.4464	10/06/2010	

FAN1	No	Hours	Date
1	0.0183	01/06/2010	
2	0.0581	05/06/2010	
3	0.0256	05/06/2010	
4	0.5772	08/06/2010	
5	0.0778	09/06/2010	
6	0.8483	10/06/2010	
7	0.8519	10/06/2010	

LAMPU 2	No	Hours	Date
1	0.0125	01/06/2010	
2	0.1069	05/06/2010	
3	0.0247	05/06/2010	
4	0.2772	09/06/2010	
5	0.3022	10/06/2010	
6	0.3022	10/06/2010	
7	0.3022	10/06/2010	

FAN 2	No	Hours	Date
1	0.0156	01/06/2010	
2	0.0436	05/06/2010	
3	0.0236	05/06/2010	
4	0.1689	09/06/2010	
5	0.0903	09/06/2010	
6	0.2084	10/06/2010	
7	0.2084	10/06/2010	

LAMPU 3	No	Hours	Date
1	0.0122	01/06/2010	
2	0.0828	05/06/2010	
3	0.0242	05/06/2010	
4	0.2639	09/06/2010	
5	0.2614	10/06/2010	
6	0.2614	10/06/2010	
7	0.2614	10/06/2010	

LAMPU 4	No	Hours	Date
1	0.0119	01/06/2010	
2	0.0772	05/06/2010	
3	0.0239	05/06/2010	
4	0.2539	09/06/2010	
5	0.2131	10/06/2010	
6	0.2131	10/06/2010	
7	0.2131	10/06/2010	

Total Time Usage					
LAMPU 1	LAMPU 2	LAMPU 3	LAMPU 4	FAN 1	FAN 2
1.5808	1.4306	1.0083	0.9222	1.8658	0.5536

Reset 1 Reset 2 Reset 3 Reset 4 Reset 5 Reset 6

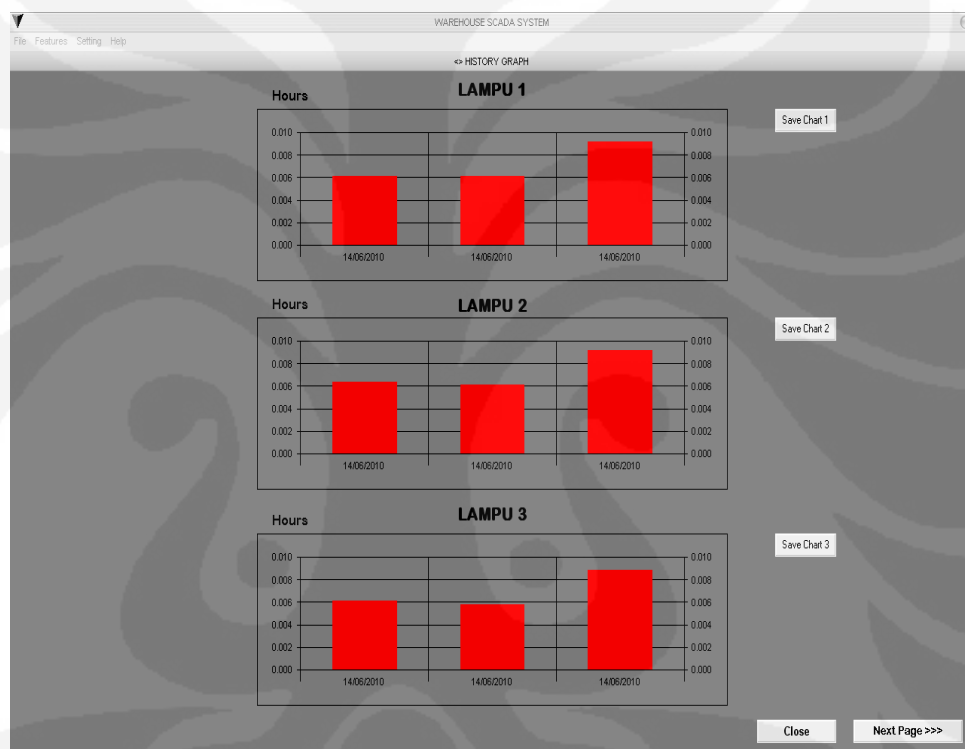
Save Total Time Usage Print

Save All Data Delete All Data Auto Save Off Close

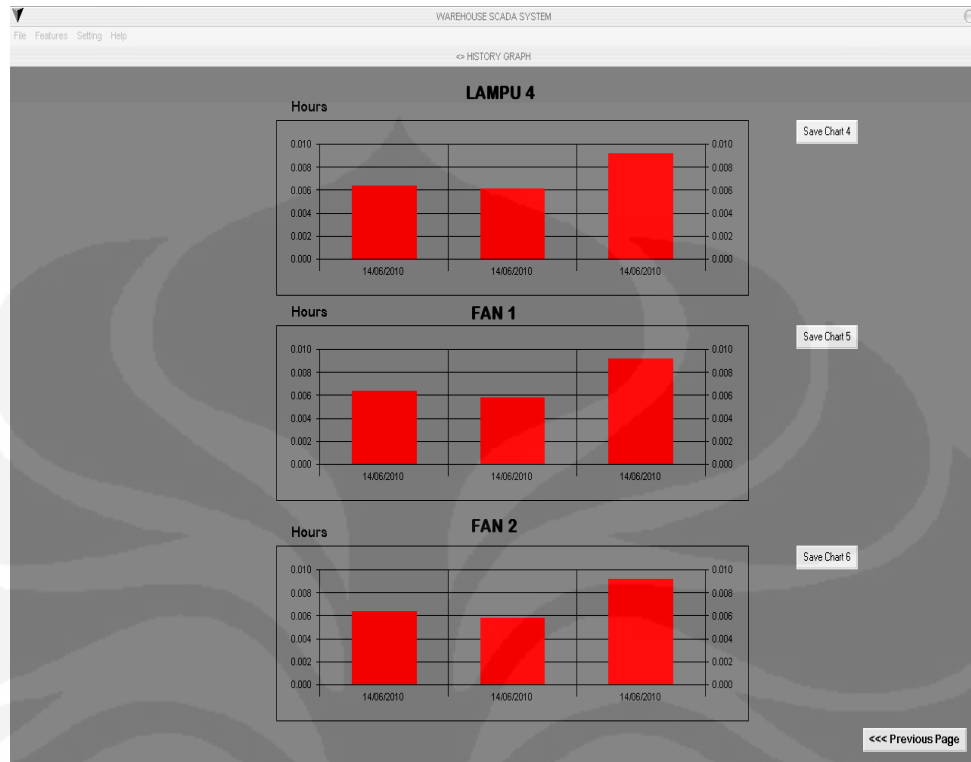
Gambar 3.15 Form Time Usage

3.3.2.4 Form History Graph

Pada *form history graph* ini akan ditampilkan grafik bar untuk lamanya pemakaian dari masing-masing unit beban. Data ini dapat digunakan untuk keperluan *maintenance* dari masing-masing unit beban tersebut. *Form history graph* ini terbagi dua bagian, yaitu pada *history graph* pertama terdapat data lampu 1, lampu 2, lampu 3 dan pada *history graph* kedua terdapat data lampu 4, *fan 1* dan *fan 2*.



Gambar 3.16 *Form History Graph 1*



Gambar 3.17 Form History Graph 2

3.3.2.5 Form Setting

Pada *form setting* ini akan ditampilkan menu untuk pengaturan waktu pemakaian maksimum dari masing-masing unit beban untuk proses *maintenance*. Data *setting time* ini digunakan untuk pemberitahuan waktu *maintenance* dari masing-masing unit beban yang ada pada *form time usage*.

Setting Time

Notice : Before Setting Time All Device Must be Turn Off

Lampu 1 : Hours Fan 1 : Hours

Lampu 2 : Hours Fan 2 : Hours

Lampu 3 : Hours

Lampu 4 : Hours

Setting Cancel

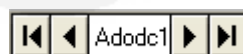
DATA LIFE TIME FOR ALL DEVICE

Lampu1	Lampu2	Lampu3	Lampu4	Fan1
8000	8000	8000	8000	10000

Gambar 3.18 Form Setting Time

3.3.3 Koneksi Database

Database adalah kumpulan informasi yang disusun berdasarkan cara tertentu dan merupakan suatu kesatuan yang utuh. Dengan sistem tersebut data yang terhimpun dalam suatu *database* dapat menghasilkan informasi yang berguna. Dalam skripsi ini digunakan koneksi *database* dengan ADODC (*ActiveX Data Object Data Control*) yang dihubungkan dengan *Microsoft Access 2007*. ADO Data Control atau yang disingkat sebagai ADODC merupakan penghubung antara kontrol-kontrol pada *form* dengan *database*. Dari segi tampilan, ADODC mempunyai bentuk yang mirip dengan DAO dimana kontrol ini mempunyai tombol arah panah yang mempunyai fungsi masing-masing yaitu untuk menunjuk ke *record* paling awal, paling akhir, menunjuk secara per-*record* ke suatu *record* sebelum atau per-*record* ke suatu *record* berikutnya. *Caption* dari kontrol ADODC secara *default* tertera dengan tulisan “Adodc1”.



Gambar 3.19 ADODC

BAB 4

PENGUJIAN SISTEM

Dalam bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Pengujian ini meliputi pengujian software HMI SCADA yaitu untuk mode *local* dan *remote*. Serta pengujian sensor pendeteksi asap (*Smoke Detector*) dan sensor gerak (*Motion Sensor*). Kemudian pengujian program-program yang ada dalam software HMI SCADA.

4.1 Pengujian mode *Local*

Pada mode ini dilakukan pengujian dengan menekan tombol *push button* yang ada pada miniatur *warehouse*, lalu diuji berapa lama waktu yang dibutuhkan hingga lampu pada HMI SCADA pada komputer menyala. Deskripsi kerja pada mode *local* ini yaitu ketika tombol *push button* pada miniatur *warehouse* ditekan maka unit-unit beban (lampu dan *fan*) di HMI SCADA pada komputer akan berubah dari hijau menjadi merah atau off ke on. Dan begitu juga lampu yang ada di miniatur *warehouse* akan diuji waktu on-nya. Untuk mengetahui kondisi unit-unit beban sedang on atau off maka digunakan *Timer 32* pada pemrograman *Visual Basic* untuk fungsi *scanning* data memori (DM) yang ada pada PLC Omron CJ1M. Mengapa pengujian dilakukan dengan set interval *time Timer 32* mulai dari 3000 ms hal ini dikarenakan program HMI SCADA akan berat, disebabkan waktu *scanning* terlalu cepat.

Berikut ini data pengujian 1 yang di dapat dengan pengaturan :

- *Timer 32, Interval Time : 3000 ms*

Tabel 4.1 Hasil pengujian 1 mode *local*

Unit Beban	Waktu (s) (Ave)	
	HMI	Miniatur
Lampu 1	8.3	0.2
Lampu 2	6.7	0.2
Lampu 3	5.9	0.2
Lampu 4	5.4	0.2
Fan 1	5.9	0.2
Fan 2	3.4	0.2

Berikut ini hasil 2 pengujian dengan pengaturan :

- *Timer 32, Interval Time : 5000 ms*

Tabel 4.2 Hasil pengujian 2 mode *local*

Unit Beban	Waktu (s) (Ave)	
	HMI	Miniatur
Lampu 1	8.3	0.2
Lampu 2	7.1	0.2
Lampu 3	7.8	0.2
Lampu 4	5.9	0.2
Fan 1	6.1	0.2
Fan 2	9.5	0.2

Kemudian dari hasil pengujian tersebut program mode *local* berhasil 100% sesuai deskripsi.

4.2 Pengujian mode *Remote*

Pada mode ini dilakukan pengujian dengan menekan tombol on-off yang ada di HMI SCADA, lalu diuji berapa lama waktu yang diperlukan sampai unit-unit beban di miniatur *warehouse* menyala. Deskripsi dari mode *remote* ini yaitu ketika tombol on-off pada HMI SCADA ditekan maka unit-unit beban (lampu dan *fan*) di miniatur *warehouse* akan menyala.

Berikut ini data pengujian 1 yang di dapat dengan pengaturan :

- *Timer 32, Interval Time : 3000 ms*

Tabel 4.3 Hasil Pengujian 1 mode *remote*

Unit Beban	Waktu (s) (Ave)	
	HMI	Miniatur
Lampu 1	0.3	0.2
Lampu 2	0.2	0.2
Lampu 3	0.3	0.2
Lampu 4	0.3	0.2
Fan 1	0.2	0.2
Fan 2	0.3	0.2

Berikut ini data pengujian 2 yang di dapat dengan pengaturan :

- *Timer 32, Interval Time : 5000 ms*

Tabel 4.4 Hasil Pengujian 2 mode *remote*

Unit Beban	Waktu (s) (Ave)	
	HMI	Miniatur
Lampu 1	0.3	0.2
Lampu 2	0.2	0.2
Lampu 3	0.3	0.2
Lampu 4	0.2	0.2
Fan 1	0.3	0.2
Fan 2	0.3	0.2

Kemudian dari hasil pengujian tersebut program mode *remote* berhasil 100% sesuai deskripsi.

4.3 Pengujian program *Smoke Detector*

Pada pengujian ini akan dilakukan dengan memberi inputan kepada *smoke detector* agar bekerja, inputan ini dapat berupa asap atau menekan tombol test yang ada pada *smoke detector*. Kemudian akan diuji berapa lama waktu yang diperlukan untuk mengaktifkan alarm pada HMI SCADA ketika *smoke detector* mulai bekerja.

Berikut ini data pengujian 1 yang di dapat dengan pengaturan :

- *Timer 32, Interval Time : 3000 ms*

Tabel 4.5 Hasil pengujian 1 program *smoke detector*

Sensor	Waktu (s) (Ave)	
	HMI	Miniatur
Smoke Detector	4.0	0.2

Berikut ini data pengujian 2 yang di dapat dengan pengaturan :

- *Timer 32, Interval Time : 5000 ms*

Tabel 4.6 Hasil pengujian 2 program *smoke detector*

Sensor	Waktu (s) (Ave)	
	HMI	Miniatur
Smoke Detector	5.8	0.2

Kemudian dari hasil pengujian tersebut program *smoke detector* berhasil 100% sesuai deskripsi.

4.4 Pengujian program *Motion Sensor*

Pada pengujian ini akan dilakukan dengan memberi inputan kepada *motion sensor* agar bekerja, untuk sensor gerak ini akan diberi inputan berupa gerakan di radius jangkauan sensor. Kemudian akan diuji berapa lama waktu yang diperlukan untuk mengaktifkan alarm pada HMI SCADA ketika *smoke detector* mulai bekerja.

Berikut ini data pengujian 1 yang di dapat dengan pengaturan :

➤ *Timer 32, Interval Time : 3000 ms*

Tabel 4.7 Hasil pengujian 1 program *motion sensor*

Sensor	Waktu (s) (Ave)	
	HMI	Miniatur
Motion Sensor	1.6	0.2

Berikut ini data pengujian 2 yang di dapat dengan pengaturan :

➤ *Timer 32, Interval Time : 5000 ms*

Tabel 4.8 Hasil pengujian 2 program *motion sensor*

Sensor	Waktu (s) (Ave)	
	HMI	Miniatur
Motion Sensor	3.0	0.2

Kemudian dari hasil pengujian tersebut program *motion sensor* berhasil 100% sesuai deskripsi.

4.5 Pengujian program *Reset Alarm*

Pada pengujian ini akan dilakukan dengan menekan tombol *reset alarm* pada HMI SCADA, lalu akan diuji berapa lama waktu yang diperlukan untuk mematikan alarm pada miniatur *warehouse*.

Berikut ini data pengujian 1 yang di dapat dengan pengaturan :

➤ *Timer 32, Interval Time : 3000 ms*

Tabel 4.9 Hasil pengujian 1 program *reset alarm*

Tombol HMI	Waktu (s) (Ave)	
	HMI	Miniatur
Reset Alarm	3.6	0.2

Berikut ini data pengujian 2 yang di dapat dengan pengaturan :

- *Timer 32, Interval Time : 5000 ms*

Tabel 4.10 Hasil pengujian 2 program *reset alarm*

Tombol HMI	Waktu (s) (Ave)	
	HMI	Miniatur
Reset Alarm	3.7	0.2

Kemudian dari hasil pengujian tersebut program *reset alarm* berhasil 100% sesuai deskripsi.

4.6 Pengujian program *Reset System*

Pada pengujian program ini dilakukan dengan menekan tombol *reset alarm* pada HMI SCADA, lalu diuji barapa lama waktu yang diperlukan untuk mereset sistem yang ada di miniatur *warehouse*.

Berikut ini data pengujian 1 yang di dapat dengan pengaturan :

- *Timer 32, Interval Time : 3000 ms*

Tabel 4.11 Hasil pengujian 1 program *reset system*

Tombol HMI	Waktu (s) (Ave)	
	HMI	Miniatur
Reset System	3.8	0.2

Berikut ini data pengujian 2 yang di dapat dengan pengaturan :

- *Timer 32, Interval Time : 5000 ms*

Tabel 4.12 Hasil pengujian 2 program *reset system*

Tombol HMI	Waktu (s) (Ave)	
	HMI	Miniatur
Reset System	4.7	0.2

Kemudian dari hasil pengujian tersebut program *reset system* berhasil 100% sesuai deskripsi.

4.7 Pengujian program *Emergency*

Pada pengujian program ini dilakukan dengan menekan tombol *Emergency* pada miniatur *warehouse*, lalu diuji waktu yang diperlukan untuk menyalakan alarm pada HMI SCADA.

Berikut ini data pengujian 1 yang di dapat dengan pengaturan :

- *Timer 32*, Interval *Time* : 3000 ms

Tabel 4.13 Hasil pengujian 1 program *emergency*

Tombol pada miniatur	Waktu (s) (Ave)	
	HMI	Miniatur
Emergency	5.2	0.2

Berikut ini data pengujian 2 yang di dapat dengan pengaturan :

- *Timer 32*, Interval *Time* : 5000 ms

Tabel 4.14 Hasil pengujian 2 program *emergency*

Tombol pada miniatur	Waktu (s) (Ave)	
	HMI	Miniatur
Emergency	5.3	0.2

Kemudian dari hasil pengujian tersebut program *emergency* berhasil 100% sesuai deskripsi.

4.8 Pengujian program *Hours Meter*

Pada pengujian ini dilakukan dengan menguji waktu yang diperlukan *hours meter* sampai mulai bekerja, yaitu pada mode *local* dan *remote*.

Berikut ini data pengujian 1 yang didapat :

➤ *Timer 32, Interval Time : 3000 ms*

Tabel 4.15 Hasil pengujian 1 pada *hours meter*

Unit Beban	On Hours Meter (s)	Error
Lampu 1	7.3	0.20%
Lampu 2	7.2	0.20%
Lampu 3	7.4	0.21%
Lampu 4	5.6	0.15%
Fan 1	4.6	0.13%
Fan 2	5.8	0.16%
Average (%)		0.18%

Berikut ini data pengujian 2 yang didapat :

➤ *Timer 32, Interval Time : 5000 ms*

Tabel 4.16 Hasil pengujian 2 pada *hours meter*

Unit Beban	On Hours Meter (s)	Error
Lampu 1	9.9	0.28%
Lampu 2	9.4	0.26%
Lampu 3	5.7	0.16%
Lampu 4	12.1	0.34%
Fan 1	5.1	0.14%
Fan 2	7.9	0.22%
Average (%)		0.23%

Dari hasil pengujian di atas dapat disimpulkan semakin lama interval waktu *scanning*, maka waktu respon pada HMI SCADA semakin lama. Hal ini berpengaruh pada waktu menyala *hours meter*, dari hasil pengujian *hours meter* didapat *error* pada saat mode *local*. *Error* yang terjadi dikarenakan pada mode *local* terlebih dahulu dijalankan oleh program *scanning* pada HMI SCADA dengan membaca kondisi data memori (DM) pada PLC.

Kemudian dari semua pengujian yang telah dilakukan dapat disimpulkan bahwa pengaturan waktu interval *timer 32* mempengaruhi kinerja

dari program secara keseluruhan. Semakin lama interval waktunya, maka waktu respon pembacaan DM PLC juga semakin lama. Dari dua pengujian pengaturan interval waktu diatas dapat disimpulkan, bahwa pengaturan pertama dengan nilai 3000 ms lebih baik dibandingkan pengaturan kedua dengan nilai 5000 ms.

Pada sistem SCADA yang dirancangan ini, desain dari HMI SCADA yang dibuat merupakan desain tetap, yaitu apabila ada pengembangan dengan penambahan unit-unit beban atau sensor maka desain dari HMI SCADA harus diubah dan *source code*-nya juga harus ditambah.

BAB 5

KESIMPULAN

1. Pada pengujian mode *local* waktu yang diperlukan untuk menyalakan unit-unit beban pada HMI SCADA lebih lama dibandingkan mode *remote*. Hal ini dikarenakan pada mode *local* terlebih dahulu dijalankan oleh program *scanning* pada HMI SCADA dengan membaca kondisi data memori (DM) pada PLC.
2. *Setting* interval waktu dari *timer scanning* mempengaruhi waktu respon program HMI SCADA.
3. Semakin cepat waktu *scanning*, maka program HMI SCADA akan semakin berat dan dapat menyebabkan *hang* pada sistem HMI SCADA.
4. Desain dari HMI SCADA yang dibuat merupakan desain tetap, yaitu apabila ada pengembangan dengan penambahan unit-unit beban atau sensor maka desain dari HMI SCADA harus diubah dan *source code*-nya juga harus ditambah.

DAFTAR REFERENSI

- [1] Pengenalan tentang SCADA
<http://learnautomation.wordpress.com/2009/02/23/introduction-to-scada/>
- [2] Konsep dasar SCADA
<http://agfi.staff.ugm.ac.id/blog/index.php/2009/03/tutorial-scada-1-apa-manfaat-scada-bagi-anda/>
- [3] Setiawan, Iwan. *Programmable Logic Control (PLC) dan Teknik Perancangan Sistem Kontrol*. Yogyakarta: Penerbit Andi, 2006.
- [4] Halvorson, Michael. *Microsoft Visual Basic 6.0 Professional Step by Step*. Jakarta: Penerbit Gramedia, 2000.
- [5] Dasar-dasar *Microsoft Access*,
http://id.wikipedia.org/wiki/Microsoft_Access
- [6] Cara kerja *Motion Sensor*,
http://en.wikipedia.org/wiki/Passive_infrared_sensor
- [7] Cara kerja *Smoke Detektor*, http://en.wikipedia.org/wiki/Smoke_detector
- [8] Datasheet PLC Omron CJ1M CPU 11, CJ1W ID211, CJ1W OD211 dan Power Supply PA 202.



LAMPIRAN 1

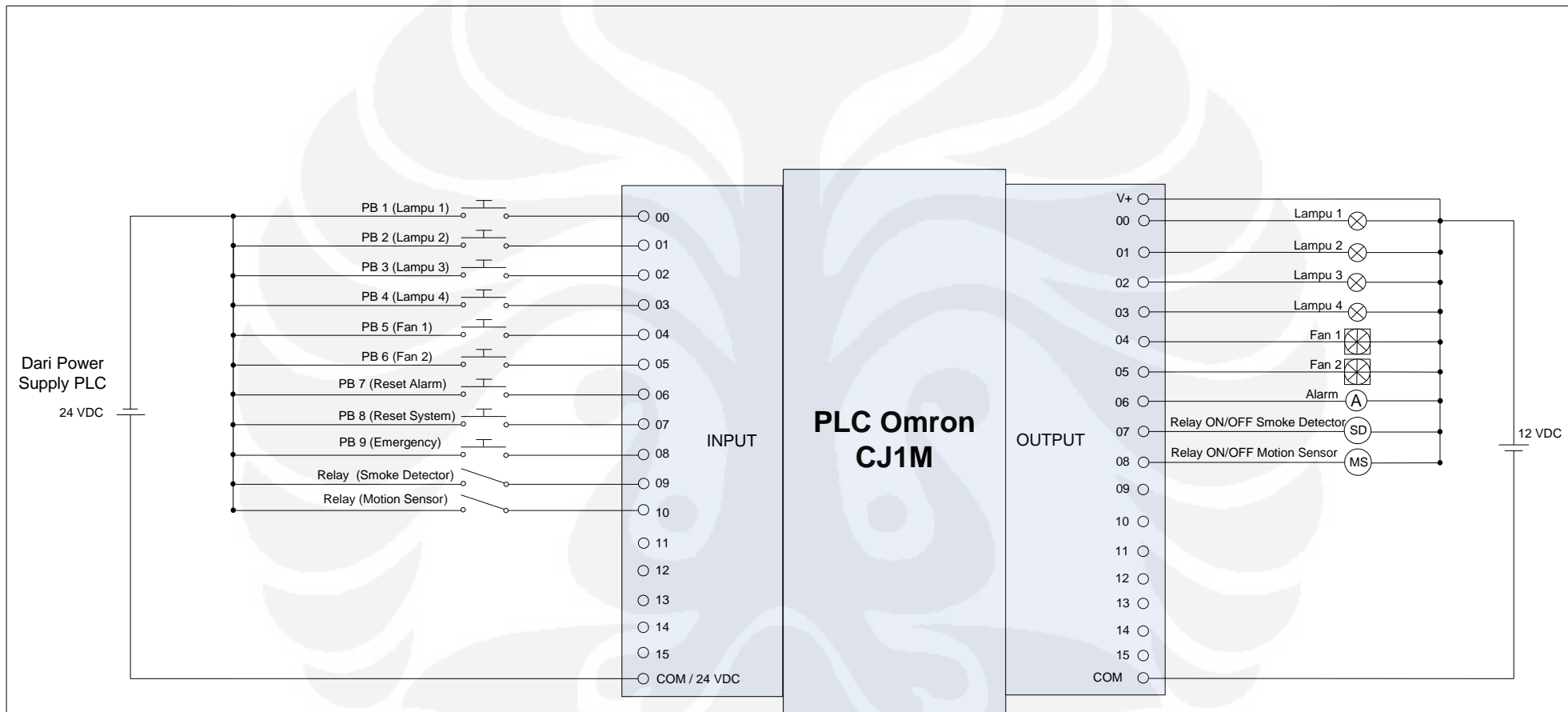
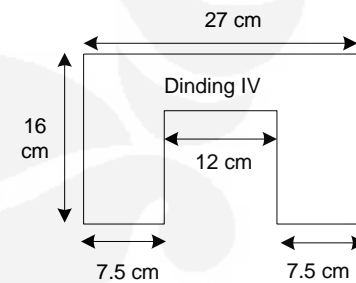
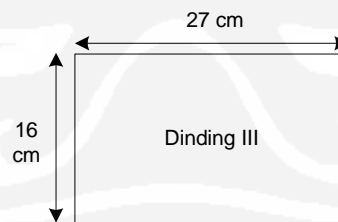
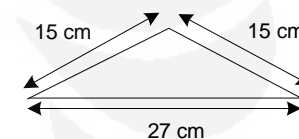
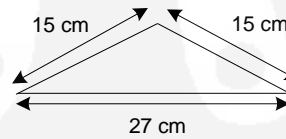
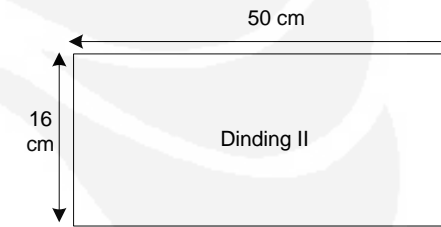
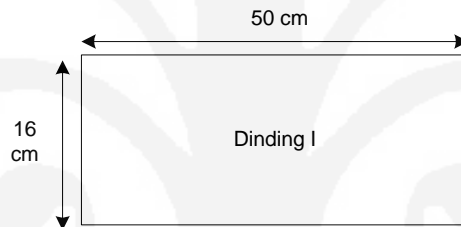
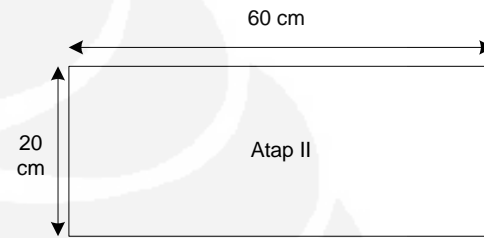
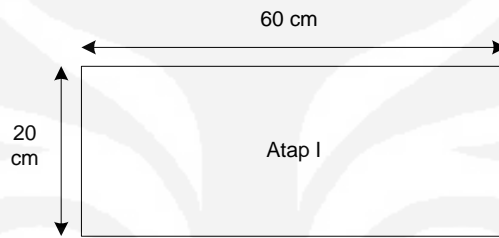
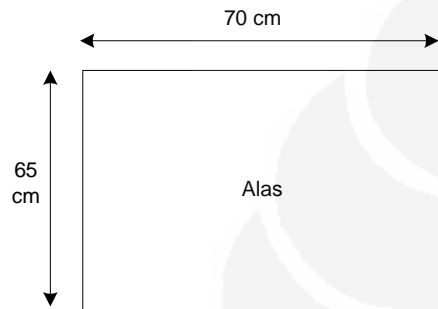
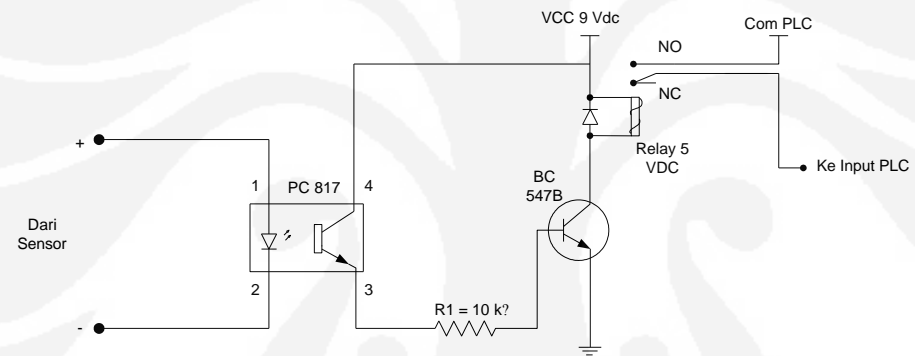


Diagram Rangkaian PLC



Dimensi *Plant Warehouse*



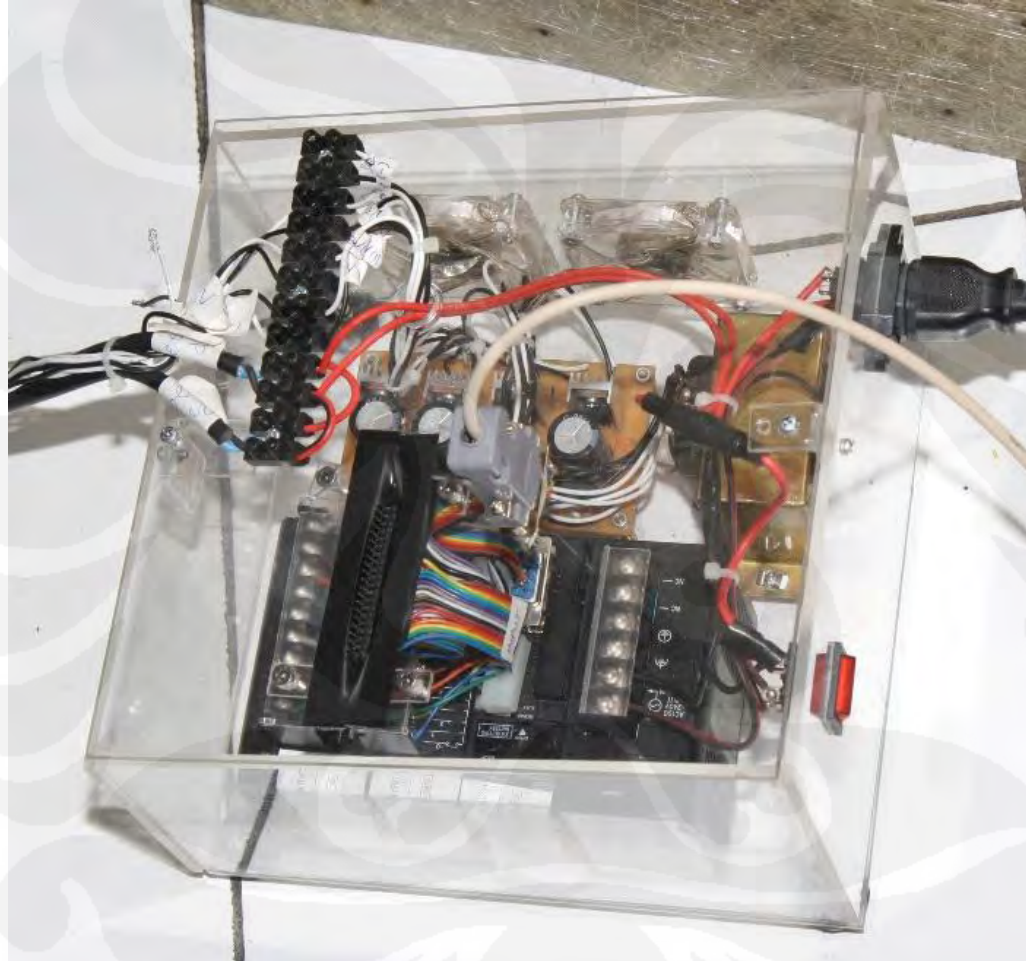
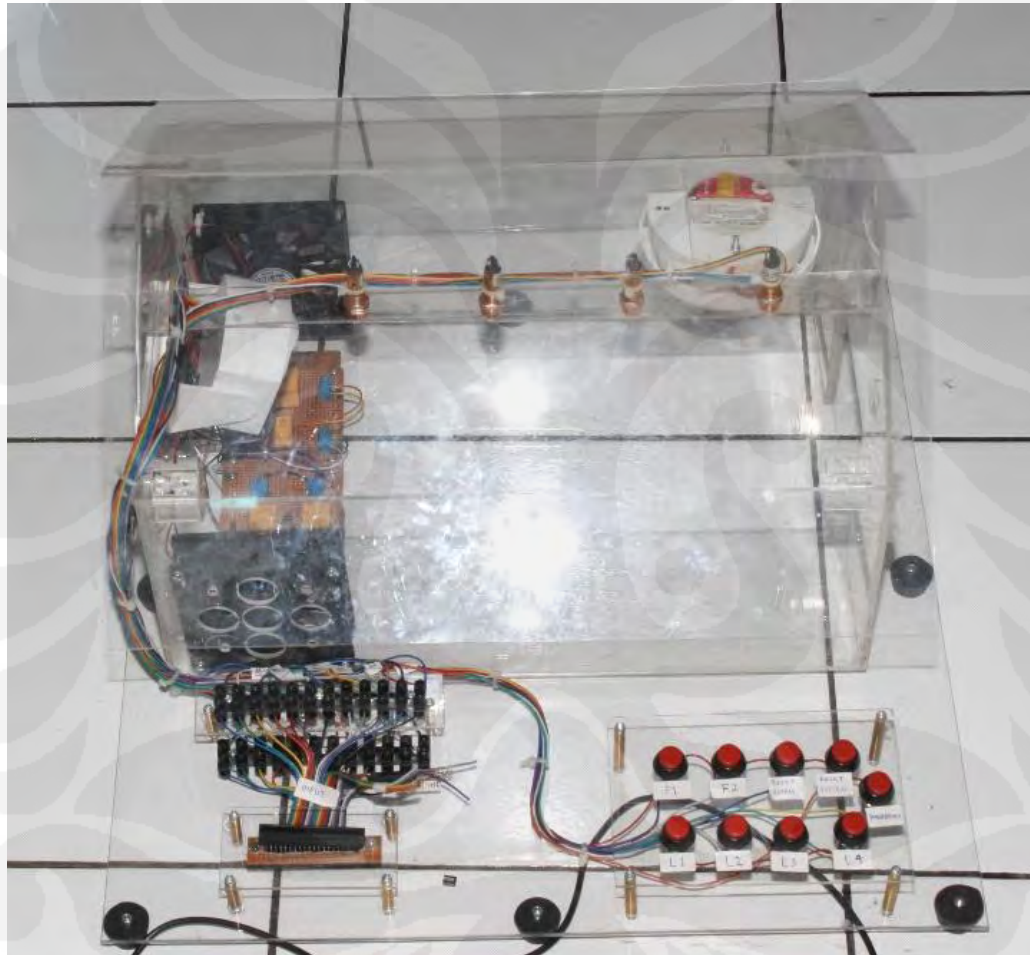


Foto Modul PLC



Perancangan sistem..., Bachtiar Salim Winarno, FT UI, 2010

Foto Miniatur *Warehouse*



LAMPIRAN 2

Form Preface :

```
Dim a As Integer
Private Sub Form_Load()
music.URL = App.Path + "\Wait.wav"
End Sub
Private Sub Timer1_Timer()
a = a + 1
pb.Value = a
DoEvents
If pb.Value = 100 Then
Timer1.Enabled = False
Set fMainform = New frmmain
Load fMainform
Unload frmpreface
fMainform.Show
End If
End Sub
Private Sub Timer2_Timer()
If lblloading.Caption = "" Then
lblloading.Caption = "Loading....Please Wait..."
Else: lblloading.Caption = ""
End If
End Sub
```

Form Main Program :


```

Dim Hari As String
Dim Dtk1, Dtk2, Dtk3, Dtk4, Dtk5, Dtk6, Dtk7, Dtk8, Dtk9, Dtk10, Dtk11, Dtk12 As Single
Dim tx3, tx4, tx5, tx6, tx7, tx8, tx9, tx10, tx11, tx12, tx13, tx14 As Single

Dim sql

Private Sub Form_Load()
    BukaDB

    Adodc1.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &
    "\Data.mdb"

    Adodc1.RecordSource = "data"

    Adodc1.Refresh

    Set DataGrid1.DataSource = Adodc1

    DataGrid1.Refresh

    AHari = Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")

    SHari = AHari(Abs(Weekday(Date) - 1))

    With StatusBar1
        .Panels(2).Text = ".....LOCAL SYSTEM ....."
        .Panels(3).Text = "" & SHari & ", " & Format(Date, "dd/mm/yyyy")
    End With

    Hari = SHari

    music2.URL = App.Path + "\Welcome.wav"

    lampu1.Enabled = False
    lampu2.Enabled = False
    lampu3.Enabled = False
    lampu4.Enabled = False

    fan1.Enabled = False
    fan2.Enabled = False
    setting.Enabled = False

    CIO_SD

End Sub

Private Sub report1()
    report.DataMember = "Command1"

End Sub

```

```

Private Sub mnulifetime_Click()
Call SetParent(frmused.hWnd, Me.hWnd)
frmused.Show
End Sub

Private Sub mnugraph_Click()
Dim rs1 As ADODB.Recordset
Set rs1 = New ADODB.Recordset
Dim rs2 As ADODB.Recordset
Set rs2 = New ADODB.Recordset
Dim rs3 As ADODB.Recordset
Set rs3 = New ADODB.Recordset
sql = "SELECT Hours FROM Lampu1"
rs1.Open sql, cn
sql = "SELECT Hours FROM Lampu2"
rs2.Open sql, cn
sql = "SELECT Hours FROM Lampu3"
rs3.Open sql, cn
If rs1.RecordCount = 0 Or rs2.RecordCount = 0 Or rs3.RecordCount = 0 Then
MsgBox "Some data Still Empty", vbYes, "Confirmation"
Else
Call SetParent(frmgrafik.hWnd, Me.hWnd)
frmgrafik.Visible = True
End If
End Sub

Private Sub mnumain_Click()
Call SetParent(frmsetting.hWnd, Me.hWnd)
frmsetting.Show
End Sub

Private Sub hapus()
BukaDB
Set rs = New ADODB.Recordset

```

```

sql = "DELETE * FROM data"

rs.Open sql, cn

Adodc1.Refresh

DataGrid1.Refresh

End Sub

Private Sub close_Click()

Dim out As Integer

out = MsgBox("Are You Sure To Exit?All Data Will Be Lost!!! ", vbQuestion + vbOKCancel,
"Confirmation")

If out = vbOK Then

hapus

DoEvents

End

End If

End Sub

Private Sub CIO_local() Local System

SYSMAC_C1.CIO(10) = 1

SYSMAC_C1.CIO(10) = 0

End Sub

Private Sub CIO_remote() Remote System

SYSMAC_C1.CIO(11) = 1

SYSMAC_C1.CIO(11) = 0

End Sub

Private Sub CIO_lampu1() Lampu 1

SYSMAC_C1.CIO(12) = 1

SYSMAC_C1.CIO(12) = 0

End Sub

Private Sub CIO_lampu2() Lampu 2

SYSMAC_C1.CIO(13) = 1

SYSMAC_C1.CIO(13) = 0

End Sub

Private Sub CIO_lampu3() Lampu 3

```

```
SYSMAC_C1.CIO(14) = 1
SYSMAC_C1.CIO(14) = 0
End Sub
Private Sub CIO_lampu4() 'Lampu 4
SYSMAC_C1.CIO(15) = 1
SYSMAC_C1.CIO(15) = 0
End Sub
Private Sub CIO_fan1() 'Fan 1
SYSMAC_C1.CIO(16) = 1
SYSMAC_C1.CIO(16) = 0
End Sub
Private Sub CIO_fan2() 'Fan 2
SYSMAC_C1.CIO(17) = 1
SYSMAC_C1.CIO(17) = 0
End Sub
Private Sub CIO_resetalarm() 'Reset Alarm
SYSMAC_C1.CIO(18) = 1
SYSMAC_C1.CIO(18) = 0
End Sub
Private Sub CIO_SD() 'ON/OFF Smoke Detector
SYSMAC_C1.CIO(19) = 1
SYSMAC_C1.CIO(19) = 0
End Sub
Private Sub CIO_MS() 'ON/OFF Motion Sensor
SYSMAC_C1.CIO(20) = 1
SYSMAC_C1.CIO(20) = 0
End Sub
Private Sub CIO_resetsystem() 'Reset System
SYSMAC_C1.CIO(21) = 1
SYSMAC_C1.CIO(21) = 0
End Sub
```

```
Private Sub Prosedur_Readstatus()
'Status Lampu 1
If SYSMAC_C1.DM(0) = 1 Then
lampu1.Tag = 1
lampu1.Picture = lampu1.DownPicture
lampu1off.Visible = False
lampu1on.Visible = True
Timer20.Enabled = True
Timer26.Enabled = True
ElseIf SYSMAC_C1.DM(0) = 2 Then
lampu1.Tag = 0
lampu1.Picture = Picture1
lampu1on.Visible = False
lampu1off.Visible = True
Timer20.Enabled = False
Timer26.Enabled = False
End If
'Status Lampu 2
If SYSMAC_C1.DM(1) = 1 Then
lampu2.Tag = 1
lampu2.Picture = lampu2.DownPicture
lampu2off.Visible = False
lampu2on.Visible = True
Timer21.Enabled = True
Timer27.Enabled = True
ElseIf SYSMAC_C1.DM(1) = 2 Then
lampu2.Tag = 0
lampu2.Picture = Picture1
lampu2on.Visible = False
lampu2off.Visible = True
```

```
Timer21.Enabled = False
Timer27.Enabled = False
End If
'Status Lampu 3
If SYSMAC_C1.DM(2) = 1 Then
lampu3.Tag = 1
lampu3.Picture = lampu3.DownPicture
DoEvents
lampu3off.Visible = False
lampu3on.Visible = True
Timer22.Enabled = True
Timer28.Enabled = True
ElseIf SYSMAC_C1.DM(2) = 2 Then
lampu3.Tag = 0
lampu3.Picture = Picture1
lampu3on.Visible = False
lampu3off.Visible = True
Timer22.Enabled = False
Timer28.Enabled = False
End If
'Status Lampu 4
If SYSMAC_C1.DM(3) = 1 Then
lampu4.Tag = 1
lampu4.Picture = lampu4.DownPicture
lampu4off.Visible = False
lampu4on.Visible = True
Timer23.Enabled = True
Timer29.Enabled = True
ElseIf SYSMAC_C1.DM(3) = 2 Then
lampu4.Tag = 0
lampu4.Picture = Picture1
```

```
lampu4on.Visible = False
lampu4off.Visible = True
Timer23.Enabled = False
Timer29.Enabled = False
End If

'Status Fan 1
If SYSMAC_C1.DM(4) = 1 Then
fan1.Tag = 1
fan1.Picture = fan1.DownPicture
fan1off.Visible = False
fan1on.Visible = True
Timer24.Enabled = True
Timer30.Enabled = True
ElseIf SYSMAC_C1.DM(4) = 2 Then
fan1.Tag = 0
fan1.Picture = Picture1
fan1on.Visible = False
fan1off.Visible = True
Timer24.Enabled = False
Timer30.Enabled = False
End If

'Status Fan 2
If SYSMAC_C1.DM(5) = 1 Then
fan2.Tag = 1
fan2.Picture = fan2.DownPicture
fan2off.Visible = False
fan2on.Visible = True
Timer25.Enabled = True
Timer31.Enabled = True
ElseIf SYSMAC_C1.DM(5) = 2 Then
fan2.Tag = 0
```

```
fan2.Picture = Picture1
fan2on.Visible = False
fan2off.Visible = True
Timer25.Enabled = False
Timer31.Enabled = False
End If
'Status ON/OFF Smoke Detector
If SYSMAC_C1.DM(6) = 1 Then
cmdsd.Tag = 1
smokedecon.Visible = True
smokedecoff.Visible = False
lblsd.Caption = "Smoke Detector ON"
ElseIf SYSMAC_C1.DM(6) = 2 Then
cmdsd.Tag = 0
smokedecon.Visible = False
smokedecoff.Visible = True
lblsd.Caption = "Smoke Detector OFF"
End If
'Status ON/OFF Motion Sensor
If SYSMAC_C1.DM(7) = 1 Then
cmdsecurity.Tag = 1
sensor1off.Visible = False
sensor1on.Visible = True
lblms.Caption = "Motion Sensor ON"
ElseIf SYSMAC_C1.DM(7) = 2 Then
cmdsecurity.Tag = 0
sensor1off.Visible = True
sensor1on.Visible = False
lblms.Caption = "Motion Sensor OFF"
End If
'Alarm Smoke Detector
```



```
If SYSMAC_C1.DM(8) = 1 Then
Timer6.Enabled = True
music.URL = App.Path + "\alarm.mp3"
Timer12.Enabled = True
lokasiString = "Alarm SD On"
waktuString = Format(Date, "dd/mm/yyyy") & " " & Format(Now, "hh:mm")
BukaDB
save_Data
Adodc1.Refresh
DataGrid1.Refresh
SYSMAC_C1.DM(8) = 0
ElseIf SYSMAC_C1.DM(8) = 2 Then
lblwarning.Caption = ""
Timer6.Enabled = False
Timer12.Enabled = False
music.Controls.stop
SYSMAC_C1.DM(8) = 0
End If
'Alarm Motion Sensor
If SYSMAC_C1.DM(9) = 1 Then
Timer19.Enabled = True
music.URL = App.Path + "\alarm.mp3"
Timer12.Enabled = True
lokasiString = "Alarm MS On"
waktuString = Format(Date, "dd/mm/yyyy") & " " & Format(Now, "hh:mm")
BukaDB
save_Data
Adodc1.Refresh
DataGrid1.Refresh
SYSMAC_C1.DM(9) = 0
ElseIf SYSMAC_C1.DM(9) = 2 Then
```

```

lblwarning.Caption = ""

Timer19.Enabled = False

Timer12.Enabled = False

music.Controls.stop

SYSMAC_C1.DM(9) = 0

End If

'Alarm Emergency
If SYSMAC_C1.DM(10) = 1 Then

Timer13.Enabled = True

music.URL = App.Path + "\alarm.mp3"

Timer12.Enabled = True

lokasiString = "Emergency On"

waktuString = Format(Date, "dd/mm/yyyy") & " " & Format(Now, "hh:mm")

BukaDB

save_Data

Adodc1.Refresh

DataGrid1.Refresh

SYSMAC_C1.DM(10) = 0

ElseIf SYSMAC_C1.DM(10) = 2 Then

lblwarning.Caption = ""

Timer13.Enabled = False

Timer12.Enabled = False

music.Controls.stop

SYSMAC_C1.DM(10) = 0

End If

End Sub

Private Sub cmdsd_Click() 'ON/OFF Smoke Detector

If cmdsd.Tag = 0 Then

cmdsd.Tag = 1

```

```
smokedecon.Visible = True
smokedecoff.Visible = False
lblsd.Caption = "Smoke Detector ON"
CIO_SD
SYSMAC_C1.DM(6) = 1
Else
cmdsd.Tag = 0
smokedecon.Visible = False
smokedecoff.Visible = True
lblsd.Caption = "Smoke Detector OFF"
CIO_SD
SYSMAC_C1.DM(6) = 2
End If
End Sub

Private Sub cmdsecurity_Click() 'ON/OFF Motion Sensor
If cmdsecurity.Tag = 0 Then
cmdsecurity.Tag = 1
sensor1off.Visible = False
sensor1on.Visible = True
lblms.Caption = "Motion Sensor ON"
CIO_MS
SYSMAC_C1.DM(7) = 1
Else
cmdsecurity.Tag = 0
sensor1off.Visible = True
sensor1on.Visible = False
lblms.Caption = "Motion Sensor OFF"
CIO_MS
SYSMAC_C1.DM(7) = 2
End If
End Sub
```

```
Private Sub cmdalarm_Click() 'Reset Alarm
CIO_resetalarm
End Sub

Private Sub cmdsystem_Click() 'Reset System
CIO_resetsystem
locrem.Tag = 0
locrem.Picture = Picture3
lbldisplay.Caption = "LOCAL SYSTEM"
Timer3.Enabled = True
Timer7.Enabled = True
Timer5.Enabled = False
Timer4.Enabled = False
lampu1.Enabled = False
lampu2.Enabled = False
lampu3.Enabled = False
lampu4.Enabled = False
fan1.Enabled = False
fan2.Enabled = False
setting.Enabled = False
Text1.Text = ""
Text2.Text = ""
Timer20.Enabled = False
Timer26.Enabled = False
Timer21.Enabled = False
Timer27.Enabled = False
Timer22.Enabled = False
Timer28.Enabled = False
Timer23.Enabled = False
Timer29.Enabled = False
Timer24.Enabled = False
Timer30.Enabled = False
```

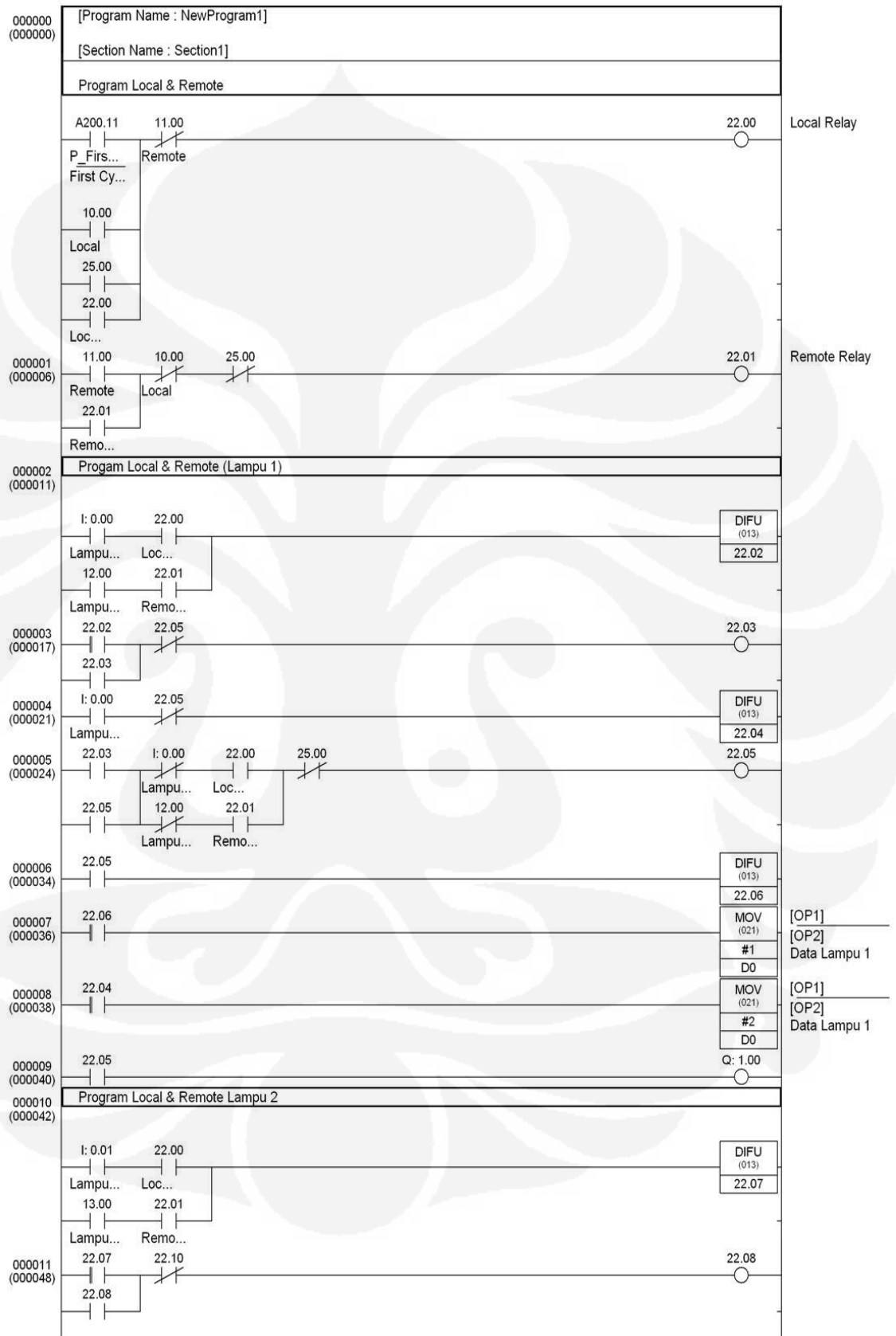
```
Timer25.Enabled = False
Timer31.Enabled = False
End Sub
Private Sub lampu1_Click()
If lampu1.Tag = 0 Then
CIO_lampu1
lampu1.Tag = 1
lampu1.Picture = lampu1.DownPicture
lampu1off.Visible = False
lampu1on.Visible = True
DoEvents
Timer20.Enabled = True
Timer26.Enabled = True
SYSMAC_C1.DM(0) = 1
Else
CIO_lampu1
lampu1.Tag = 0
lampu1.Picture = Picture1
lampu1on.Visible = False
lampu1off.Visible = True
Timer20.Enabled = False
Timer26.Enabled = False
SYSMAC_C1.DM(0) = 2
End If
End Sub
Private Sub lampu2_Click()
If lampu2.Tag = 0 Then
CIO_lampu2
lampu2.Tag = 1
lampu2.Picture = lampu2.DownPicture
lampu2off.Visible = False
```

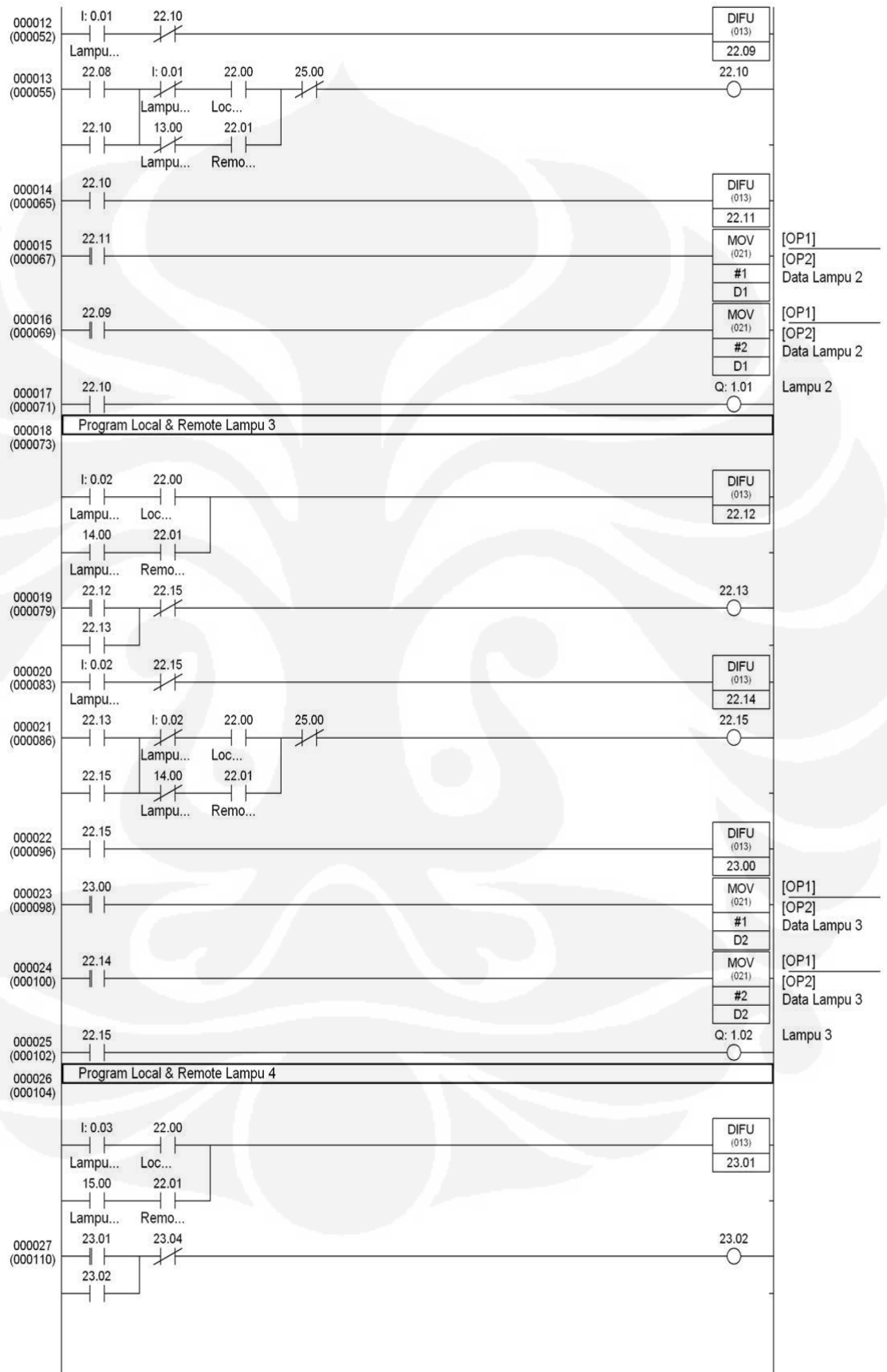
```
lampu2on.Visible = True
DoEvents
Timer21.Enabled = True
Timer27.Enabled = True
SYSMAC_C1.DM(1) = 1
Else
CIO_lampu2
lampu2.Tag = 0
lampu2.Picture = Picture1
lampu2on.Visible = False
lampu2off.Visible = True
Timer21.Enabled = False
Timer27.Enabled = False
SYSMAC_C1.DM(1) = 2
End If
End Sub
Private Sub lampu3_Click()
If lampu3.Tag = 0 Then
CIO_lampu3
lampu3.Tag = 1
lampu3.Picture = lampu3.DownPicture
lampu3off.Visible = False
lampu3on.Visible = True
DoEvents
Timer22.Enabled = True
Timer28.Enabled = True
SYSMAC_C1.DM(2) = 1
Else
CIO_lampu3
lampu3.Tag = 0
lampu3.Picture = Picture1
```

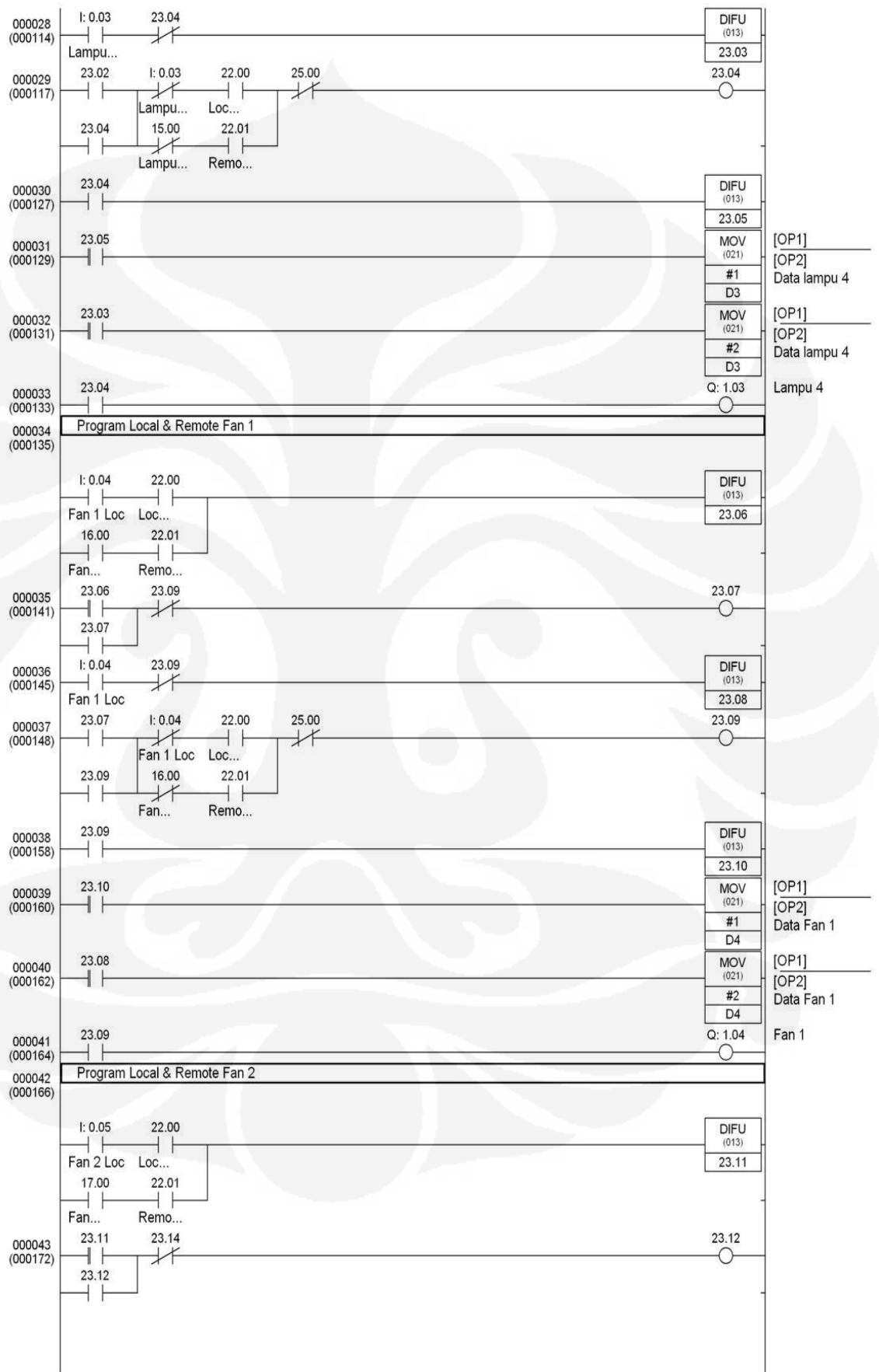
```
lampu3on.Visible = False
lampu3off.Visible = True
Timer22.Enabled = False
Timer28.Enabled = False
SYSMAC_C1.DM(2) = 2
End If
End Sub
Private Sub lampu4_Click()
If lampu4.Tag = 0 Then
CIO_lampu4
lampu4.Tag = 1
lampu4.Picture = lampu4.DownPicture
lampu4off.Visible = False
lampu4on.Visible = True
DoEvents
Timer23.Enabled = True
Timer29.Enabled = True
SYSMAC_C1.DM(3) = 1
Else
CIO_lampu4
lampu4.Tag = 0
lampu4.Picture = Picture1
lampu4on.Visible = False
lampu4off.Visible = True
Timer23.Enabled = False
Timer29.Enabled = False
SYSMAC_C1.DM(3) = 2
End If
End Sub
```

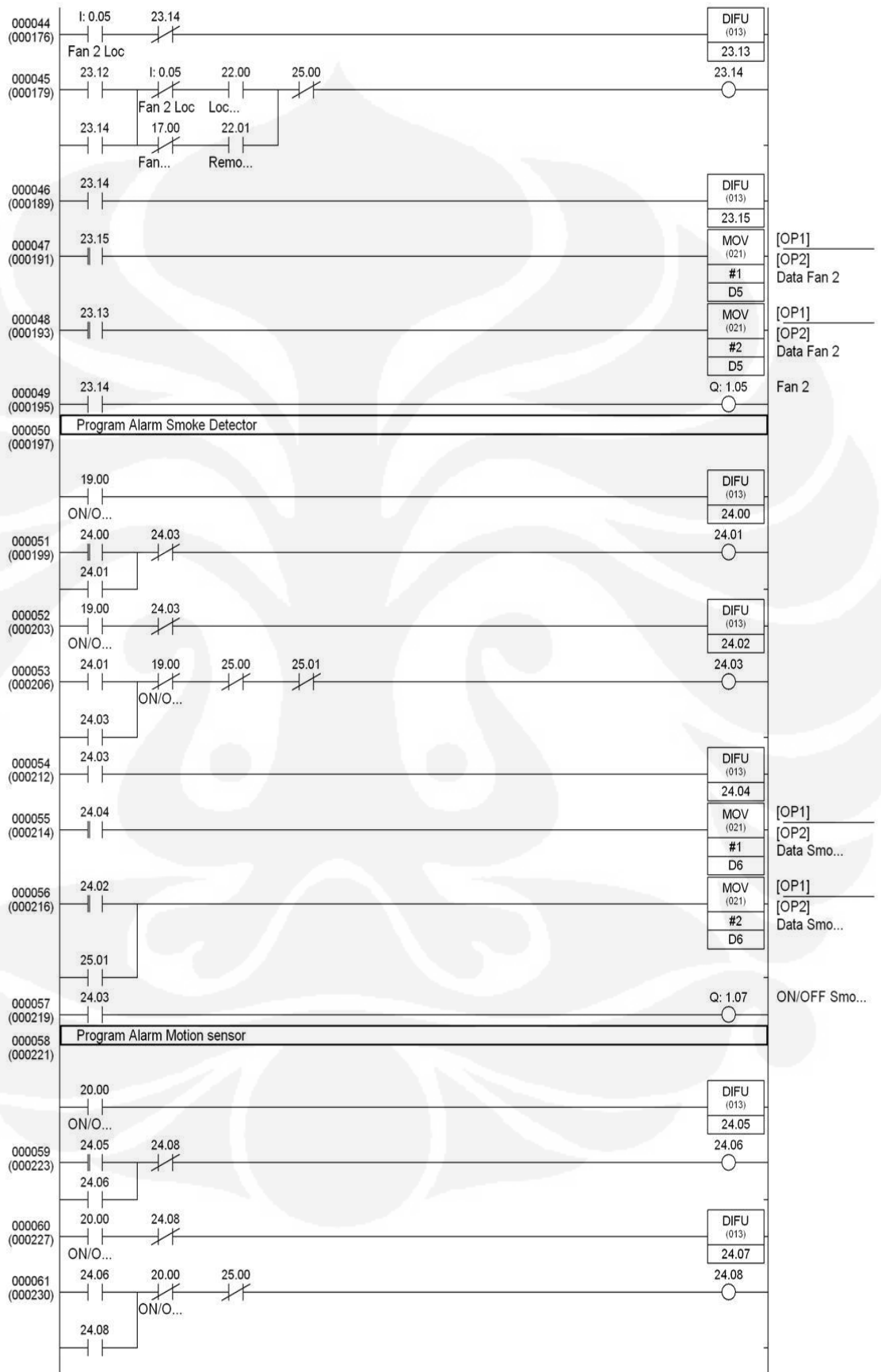
LAMPIRAN 3

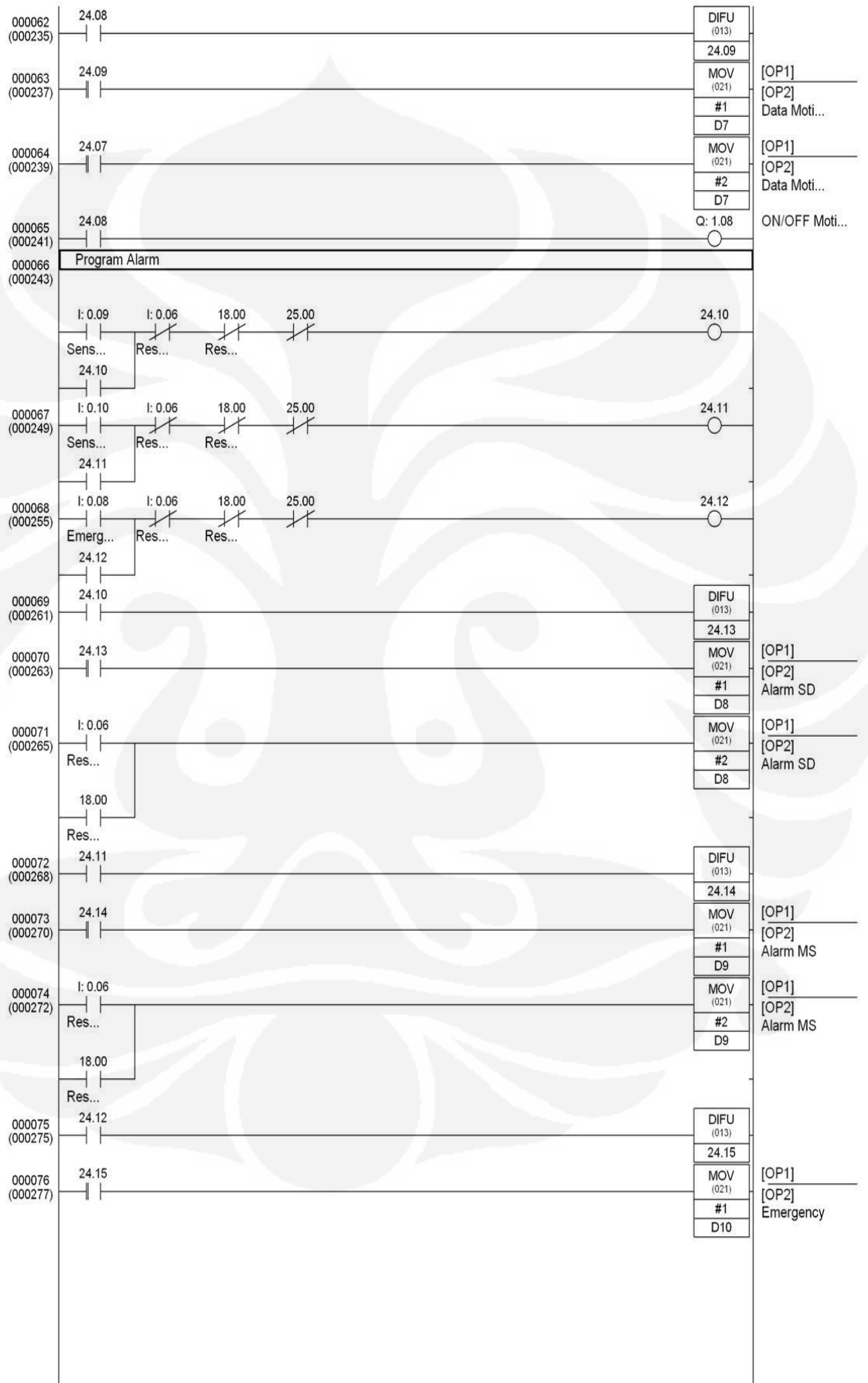


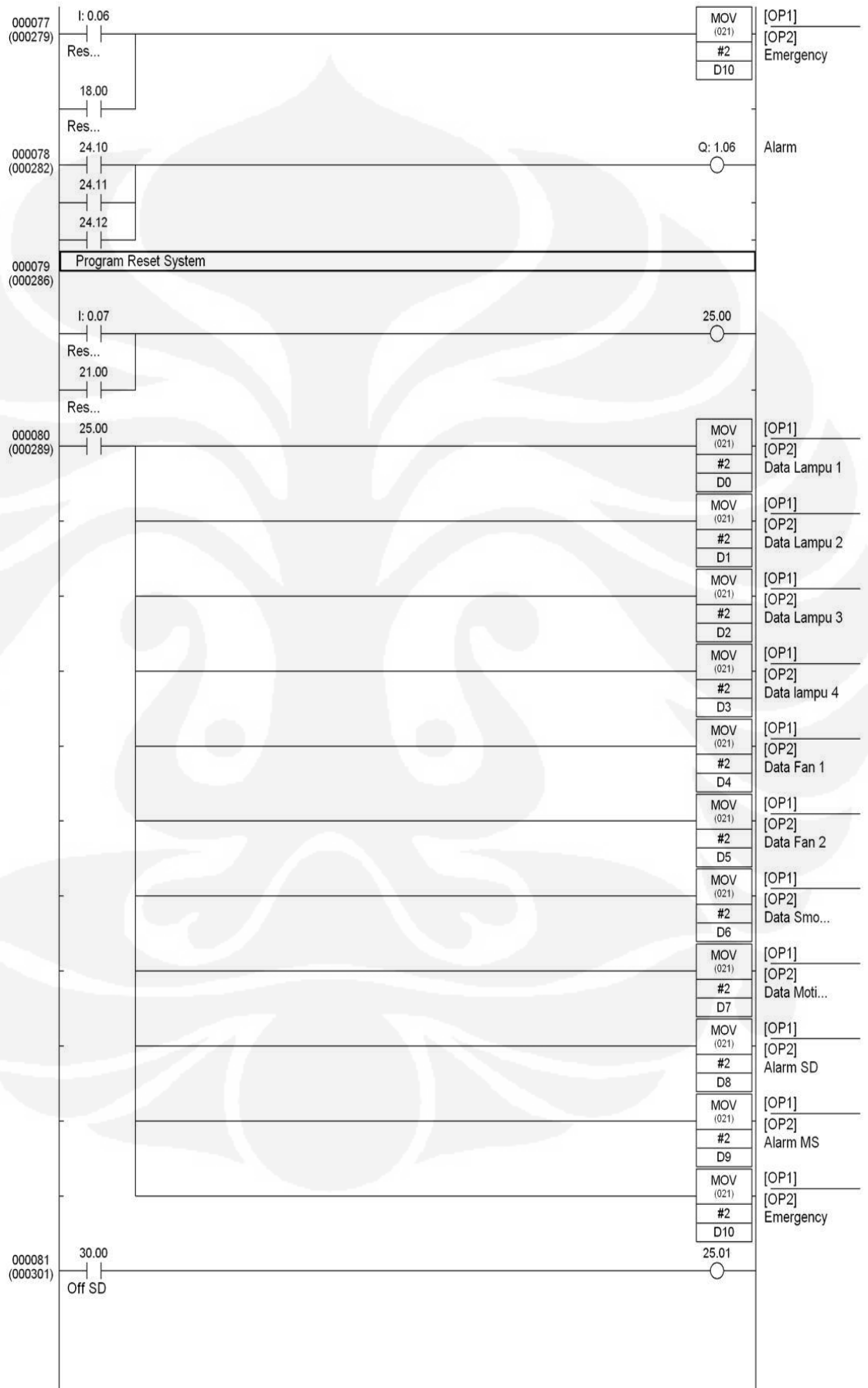














LAMPIRAN 4

SYSMAC CJ-series Input Units

CJ1W-ID/IA

PLC C2W000111 08 E A 1

A Wide Range of Basic Input Units for High Speed Input and Different Applications

- Receive ON/OFF signals from external devices into the PLC System to update I/O memory in the CPU Unit.
- New high-speed input models CJ1W-ID212 and CJ1W-ID233 are now available. These units can help to increase system throughput.



CJ1W-ID212 **NEW**



CJ1W-ID233 **NEW**

Features

- High-speed input models are available, meeting versatile applications.
ON Response Time: 15 μ s, OFF Response Time: 90 μ s
 - Use 24-VDC, 100-VAC, and 200-VAC models to connect to devices with different types of outputs.
 - The 24-VDC models can be connected to devices with either NPN or PNP outputs. There is no need to select the polarity. (See note 1.)
 - A digital filter in the Unit can be set from 0 to 32 ms to reduce the influence of external noise.
 - Either a Fujitsu or MIL connector interface can be used. (See note 2.)
 - Several models of Terminal Block Conversion Units are available, making it easy to connect to external devices.
- Note:** 1. The same polarity is used for the same common.
2. For models with 32 or 64 inputs.






Ordering Information

International Standards

- The standards are abbreviated as follows: U: UL, U1: UL (Class 1 Division 2 Products for Hazardous Locations), C: CSA, UC: cULus, UC1: cULus (Class 1 Division 2 Products for Hazardous Locations), CU: cUL, N: NK, L: Lloyd, and CE: EC Directives.
- Contact your OMRON representative for further details and applicable conditions for these standards.

Input Units

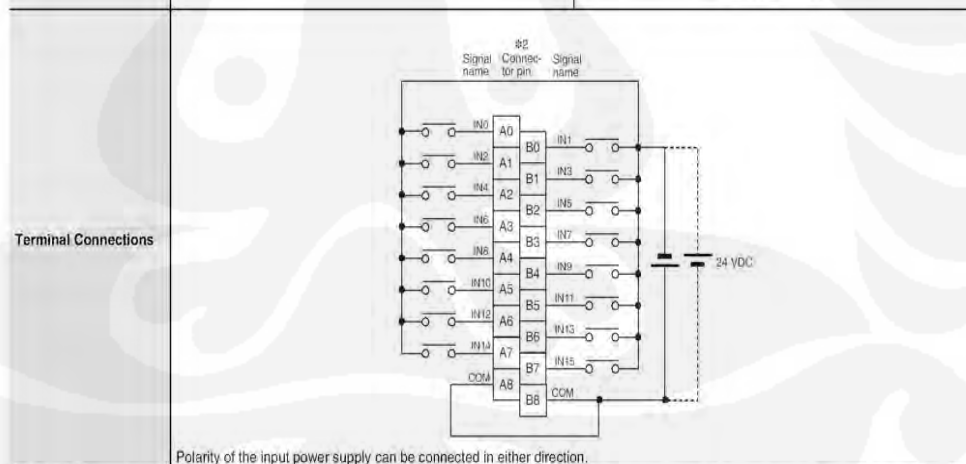
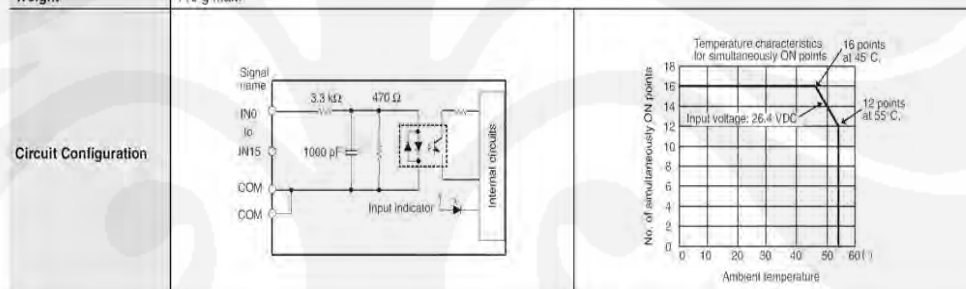
Unit type	Product name	Specifications					Current consumption (A)		Model	Standards	
		I/O points	Input voltage and current	Commons	External connection	No. of words allocated	5 V	24 V			
CJ1 Basic I/O Units	DC Input Units	8 inputs	12 to 24 VDC, 10 mA	Independent contacts	Removable terminal block	1 word	0.09	–	CJ1W-ID201	UC1, N, L, CE	
		16 inputs	24 VDC, 7 mA	16 points, 1 common	Removable terminal block	1 word	0.08	–	CJ1W-ID211		
		16 inputs (High speed)	24 VDC, 7 mA	16 points, 1 common	Removable terminal block	1 word	0.13	–	^{NEW} CJ1W-ID212	N, L, CE	
		32 inputs	24 VDC, 4.1 mA	16 points, 1 common	Fujitsu connector	2 words	0.09	–	CJ1W-ID231		
		32 inputs	24 VDC, 4.1 mA	16 points, 1 common	MIL connector	2 words	0.09	–	CJ1W-ID232	UC1, N, L, CE	
		32 inputs (High speed)	24 VDC, 4.1 mA	16 points, 1 common	MIL connector	2 words	0.20	–	^{NEW} CJ1W-ID233		
		64 inputs	24 VDC, 4.1 mA	16 points, 1 common	Fujitsu connector	4 words	0.09	–	CJ1W-ID261		
		64 inputs	24 VDC, 4.1 mA	16 points, 1 common	MIL connector	4 words	0.09	–	CJ1W-ID262		
	AC Input Units		8 inputs	200 to 24 VAC, 10 mA (200 V, 50 Hz)	8 points, 1 common	Removable Terminal Block	1 words	0.08	–	CJ1W-IA201	UC1, N, L, CE
			16 inputs	100 to 120 VAC, 7 mA (100 V, 50 Hz)	16 points, 1 common	Removable Terminal Block	1 words	0.09	–	CJ1W-IA111	

Accessories

Connectors are not included for models with connectors. Either use one of the applicable connector listed below or use an applicable Connector-Terminal Block Conversion Unit or I/O Relay Terminal. For details on wiring methods, refer to *External Interface*.

CJ1W-ID211 DC Input Unit (24 VDC, 16 Points)

Name	16-point DC Input Unit with Terminal Block
Model	CJ1W-ID211
Rated Input Voltage	24 VDC
Rated Input Voltage Range	20.4 to 26.4 VDC
Input Impedance	3.3 k Ω
Input Current	7 mA typical (at 24 VDC)
ON Voltage/ON Current	14.4 VDC min./3 mA min.
OFF Voltage/OFF Current	5 VDC max./1 mA max.
ON Response Time	8.0 ms max. (Can be set to between 0 and 32 ms in the PLC Setup.) *1
OFF Response Time	8.0 ms max. (Can be set to between 0 and 32 ms using PLC Setup) *1
Number of Circuits	16 (16 points/common, 1 circuit)
Number of Simultaneously ON Points	100% simultaneously ON (at 24 VDC) (Refer to the following illustration.)
Insulation Resistance	20 M Ω between external terminals and the GR terminal (100 VDC)
Dielectric Strength	1,000 VAC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	80 mA max.
Weight	110 g max.



*1. The ON response time will be 20 μ s maximum and OFF response time will be 400 μ s maximum even if the response time are set to 0 ms due to internal element delays.

*2. Terminal numbers A0 to A8 and B0 to B8 are used in this datasheet, but they are not printed on the Unit.

SYSMAC CJ-series Output Units

CJ1W-OC/OA/OD

C&M_CJ1W-OUTPUT_DS_E_1.1

A Wide Range of Basic Output Units for High Speed Output and Different Applications

- These Output Units receive the results of output instructions from the CPU Unit and perform ON/OFF control for external devices.
- New high-speed Output models CJ1W-OD213 and CJ1W-OD234 now available. These units can help to increase system throughput.



CJ1W-OD213 **NEW**



CJ1W-OD234 **NEW**




Features

- High-speed output models are available, meeting versatile applications.
ON Response Time: 15 μ s, OFF Response Time: 80 μ s
 - Output Units are available with any of three output types: relay contact outputs, triac outputs, or transistor outputs.
 - For transistor outputs, select from sinking outputs or sourcing outputs.
 - Output Units with load short-circuit protection are also available. *1
 - Select the best interface for each application: Fujitsu connectors or MIL connectors. *2
 - A wide variety of Connector-Terminal Block Conversion Units are available to allow you to easily wire external output devices.
- *1. The following Units have load short-circuit protection: CJ1W-OC202, CJ1W-OD204, CJ1W-OD212, and CJ1W-OD232.
*2. Available for models with 32 outputs or 64 outputs



Ordering Information

Output Units

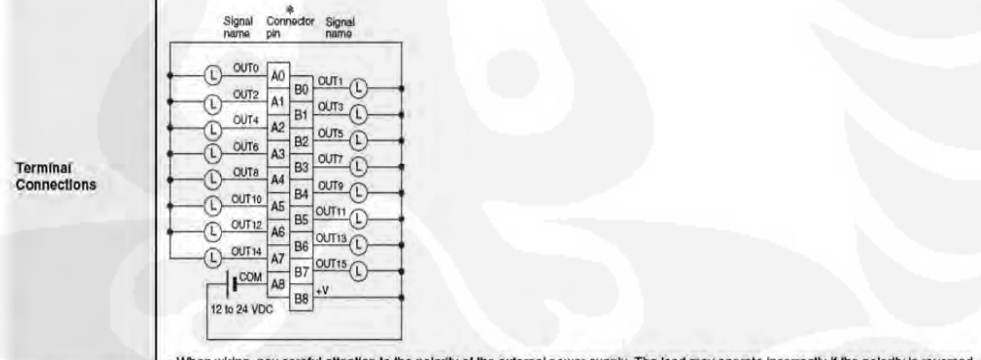
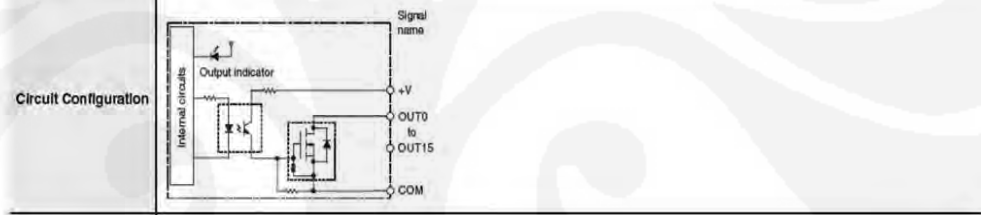
Unit type	Product name	Specifications					No. of words allocated	Current consumption (A)		Model	Standards	
		Output type	I/O points	Maximum switching capacity	Commons	External connection		5 V	24 V			
CJ1 Basic I/O Units	 Relay Contact Output Units	-	8 outputs	250 VAC/24 VDC, 2 A	Independent contacts	Removable terminal block	1 words	0.09	0.048 max.	CJ1W-OC201	UC1, N, L, CE	
		-	16 outputs	250 VAC/24 VDC, 2 A	16 points, 1 common	Removable terminal block	1 words	0.11	0.096 max.	CJ1W-OC211		
	 Triac Output Unit	-	8 outputs	250 VAC, 0.6 A	8 points, 1 common	Removable terminal block	1 words	0.22	-	CJ1W-OA201		
	 Transistor Output Units	Sinking	8 outputs	12 to 24 VDC, 2 A	4 points, 1 common	Removable terminal block	1 words	0.09	-	CJ1W-OD201		
		Sinking	8 outputs	12 to 24 VDC, 0.5 A	8 points, 1 common	Removable terminal block	1 words	0.10	-	CJ1W-OD203		
		Sinking	16 outputs	12 to 24 VDC, 0.5 A	16 points, 1 common	Removable terminal block	1 words	0.10	-	CJ1W-OD211		
		Sinking	16 outputs (High speed)	24 VDC, 0.5 A	16 points, 1 common	Removable terminal block	1 words	0.15	-	^{NEW} CJ1W-OD213		N, L, CE
		Sinking	32 outputs	12 to 24 VDC, 0.5 A	16 points, 1 common	Fujitsu connector	2 words	0.14	-	CJ1W-OD231		UC1, N, L, CE
		Sinking	32 outputs	12 to 24 VDC, 0.5 A	16 points, 1 common	MIL connector	2 words	0.14	-	CJ1W-OD233		
		Sinking	32 outputs (High speed)	24 VDC, 0.5 A	16 points, 1 common	MIL connector	2 words	0.22	-	^{NEW} CJ1W-OD234		N, L, CE
		Sinking	64 outputs	12 to 24 VDC, 0.3 A	16 points, 1 common	Fujitsu connector	4 words	0.17	-	CJ1W-OD261		UC1, N, L, CE
		Sinking	64 outputs	12 to 24 VDC, 0.3 A	16 points, 1 common	MIL connector	4 words	0.17	-	CJ1W-OD263		
		Sourcing	8 outputs	24 VDC, 2 A Short-circuit protection	4 points, 1 common	Removable terminal block	1 words	0.11	-	CJ1W-OD202		
		Sourcing	8 outputs	24 VDC, 0.5 A Short-circuit protection	8 points, 1 common	Removable terminal block	1 words	0.10	-	CJ1W-OD204		
		Sourcing	16 outputs	24 VDC, 0.5 A Short-circuit protection	16 points, 1 common	Removable terminal block	1 words	0.10	-	CJ1W-OD212		
	Sourcing	32 outputs	24 VDC, 0.5 A Short-circuit protection	16 points, 1 common	MIL connector	2 words	0.15	-	CJ1W-OD232			
	Sourcing	64 outputs	12 to 24 VDC, 0.3 A	16 points, 1 common	MIL connector	4 words	0.17	-	CJ1W-OD262			

Accessories

Connectors are not included for models with connectors. Either use one of the applicable connector listed below or use an applicable Connector-Terminal Block Conversion Unit or I/O Relay Terminal. For details on wiring methods, refer to *External Interface*.

CJ1W-OD211 Transistor Output Unit (16 Points)

Name	16-point Transistor Output Unit with Terminal Block (Sinking Outputs)
Model	CJ1W-OD211
Rated Voltage	12 to 24 VDC
Operating Load Voltage Range	10.2 to 26.4 VDC
Maximum Load Current	0.5 A/point, 5.0 A/Unit
Maximum Inrush Current	4.0 A/point, 10 ms max.
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.1 ms max.
OFF Response Time	0.8 ms max.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 VDC)
Dielectric Strength	1,000 VAC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Number of Circuits	16 (16 points/common, 1 circuit)
Internal Current Consumption	5 VDC 100 mA max.
Fuse	None
External Power Supply	10.2 to 26.4 VDC, 20 mA min.
Weight	110 g max.



* When wiring, pay careful attention to the polarity of the external power supply. The load may operate incorrectly if the polarity is reversed.
 * Terminal numbers A0 to A8 and B0 to B8 are used in this datasheet, but they are not printed on the Unit.

CJ Series Power Supply Unit

CJ1W-PA/PD

Equipped with the RUN output for checking the operation status, as well as the replacement notification function for easy maintenance.

Lineup of five models including the AC power supply (25W).



CJ1W-PA205C

CJ1W-PA202

- Stable power supply is available from the CJ-series CPU Unit to each I/O Units via the dedicated bus. The most suitable Power Supply Unit can be selected among various I/O specifications and/or options.

Features

- The replacement notification function can prevent an overflow of the system due to the power lifetime. (CJ1W-PA205C only)
 - Power Supply Unit provides wide variations according to the system scale up to maximum 25 W.
 - Power Supply Unit provides wide variations according to the power supply (100 to 240 V AC/ 24 V DC) status.
 - The output contact during operation enables you to check the CPU operation. (CJ1W-PA205R only)
 - Conforming to the SEMI-F47 Standard. *
- * AC input type, 200 V min. input.



■ Main Specifications of C.J1M CPU Units

Model	Standard CPU Units			CPU Units with Pulse I/O	
	NEW CJ1M-CPU11	CJ1M-CPU12	CJ1M-CPU13	NEW CJ1M-CPU21	CJ1M-CPU22
I/O points	160 points	320 points	640 points	160 points	320 points
Expansion Rack	Not supported.	Not supported.	One Rack	Not supported.	One Rack
Maximum number of connected Units	10 Units	10 Units	20 Units (CPU Rack; Expansion Rack: 10 Units)	10 Units	20 Units (CPU Rack; Expansion Rack: 10 Units)
Program memory capacity	5 Ksteps	10 Ksteps	20 Ksteps	5 Ksteps	10 Ksteps
Data memory capacity	32 Kwords (DM only, no EM)				
LD instruction	100 ns				
MOV instruction	0.3 μs				
Overhaul time	0.7 ms	0.5 ms		0.7 ms	0.5 ms
Pulse start time	—			63 μs (without acceleration/deceleration)	46 μs (without acceleration/deceleration)
				100 μs (with acceleration/deceleration)	70 μs (with acceleration/deceleration)
Built-in I/O	One peripheral port RS-485				
Mounting positions	Vertical				
Number of subroutines and jump instructions	256	1024		256	1024
Number of scheduled interrupts	1	2		1	2
Built-in I/Os	—			10 inputs -2 pulse outputs, 100-kHz -2 high-speed counter inputs (50-kHz phase-differential or 100-kHz single-phase inputs)	6 outputs -2 pulse outputs, 100 kHz -2 PWM outputs -1 PWM output
Built-in I/O points	—				

Warranty and Limitations of Liability

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, FOR THE PRODUCTS. ANY BUYER OR USER ACKNOWLEDGES THAT THE BUYER OR USER SHALL OBTAIN AND READ THE INSTRUCTIONS AND CAUTIONS FOR THE PRODUCTS BEFORE USING THEM. THE BUYER OR USER SHALL OBTAIN AND READ THE INSTRUCTIONS AND CAUTIONS FOR THE PRODUCTS BEFORE USING THEM. THE BUYER OR USER SHALL OBTAIN AND READ THE INSTRUCTIONS AND CAUTIONS FOR THE PRODUCTS BEFORE USING THEM. THE BUYER OR USER SHALL OBTAIN AND READ THE INSTRUCTIONS AND CAUTIONS FOR THE PRODUCTS BEFORE USING THEM.

Note: Do not use this document to operate the Unit.

OMRON Corporation
 FA Systems Division H.Q.
 6-2, Honjo-cho, Matsuyama-shi, Ehime-ken, Japan
 Tel: (81) 83-927-1811
 Fax: (81) 83-977-5685

Regional Headquarters
OMRON EUROPE B.V.
 Weismann 67-69, NL-2152 JD 'Hoddonp
 Maastricht, The Netherlands
 Tel: (31) 43 192346-RL-3000/Fax: (31) 43 192348-3188

OMRON ELECTRONICS LLC
 1 East Commerce Drive, Schaumburg, IL 60173
 U.S.A.
 Tel: (1) 815 895-7900/Fax: (1) 815 895-4568

OMRON ASIA PACIFIC PTE. LTD.
 #11-01, UJE Square,
 85 Chennongshu Avenue,
 Singapore
 Tel: (65) 9983-5301/Fax: (65) 9983-5211

Authorized Distributor:
 Note: Specifications subject to change without notice. Call No. R114.ET.02 Printed in Japan. 01P/2K

New!
 New Product News

Programmable Controllers
SYSTMAC
CJ1M
 Low-end Models

OMRON

Lineup of Low-end Models with 160 I/O Points and 5-Kstep Capacity
 CJ1M-CPU1 1/21 Added to Series
 Providing Greater Added Value for Small-scale Machines



A PLC with a 64-point Mixed I/O Unit is shown above (121.7 × 90.0 × 65.0 mm, W × H × D). (Refer to the next page for configuration details.)

ECO
 Energy Conservation
 Environmentally Conscious
 OMRON

Innovation in the Solution Age
 OMRON INDUSTRIAL AUTOMATION

CJ1M CPU Units: Small yet Powerful, and Now in More Cost-efficient Models

Lineup of Low-end CPU Units with Basic Functions Plus High Added Value
 Program capacity of 5 Ksteps and up to 180 I/O points: For small-scale machine control, yet including all the basic functions. Standard features include a Memory Card interface and RS-232C port for serial communications. CPU Units with Pulse I/O have built-in high-speed counter inputs and pulse outputs for machines.



- **Peripheral Port**
Connect a Programming Device, PT, etc.
- **RS-232C Port**
Connect a PT, Barcode Reader, Programming Device, etc.
- **Memory Card Interface**

Standard CPU Unit
CJ1M-CPU11
 180 I/O points, program capacity: 5 Ksteps



- **Peripheral Port**
Connect a Programming Device, PT, etc.
- **RS-232C port**
Connect a Barcode Reader, Programming Device, etc.
- **CPU Unit's Built-in I/O**
10 inputs (pulse inputs), 6 outputs (pulse/PWM outputs)
- **Memory Card Interface**

CPU Unit with Pulse I/O
CJ1M-CPU21
 180 I/O points, program capacity: 5 Ksteps, 16 built-in I/O points

Choose the optimum Units to suit the application.



Select the optimum PLC from a complete lineup of low- to high-end CJ-series CPU Units to match your machine application needs.



The CJ1M-CPU11/21 are supported by CX-Programmer version 3.3.

CJ1M-CPU11/21 Features

Save space
 Compact with No Backplane



Backplane-free Configuration Flexible combinations



Improve Machine Speed

Reduced tact time, improved productivity. Execute LD instructions at 100 ns and MOV instructions at 0.3 μs for higher machine speed.

The following units are also available in the following products:
 CJ1M-PAD20 Power Supply Unit
 CJ1M-CPU11 CPU Unit
 CJ1M-MD201 64-point Mixed I/O Unit

Features of the CJ1M-CPU21 CPU Unit with Pulse I/O

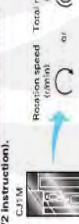
Encoder Inputs for Two Axes

100-kHz single phase, or 50-kHz phase-differential pulse inputs



Measure Rotation Data

Convert from high-speed counter input pulses to rotation speeds for total rotations (new PRVZ instruction).



Four Interrupt or Quick-response Inputs

Quick-response inputs do not require a special interrupt input unit.



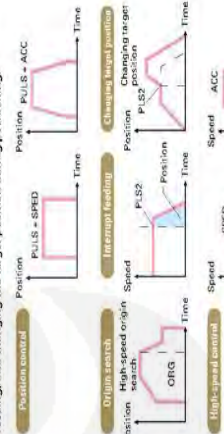
High-precision Pulse with Variable Duty Factor (PWM) Output

The duty ratio can be set in 0.1% units.



High-speed Pulse Outputs for Two Axes

Wide variety of pulse output functions, such as 100-kHz isokinetic or trapezoidal acceleration/deceleration for speed control, origin searches, high-speed origin search, high-speed origin search with high-speed feeding, and changing the target position during positioning.



Acceleration/Deceleration Suppression

Source Acceleration/Deceleration Suppression decreases vibration during high-speed positioning in equipment with low rigidity.



NEW Functions indicated as new are supported by CX-Programmer version 4.0 (available soon) or later. (Compatible with CPU Unit Ver.2.0)