



**UNIVERSITAS INDONESIA**

**APLIKASI SCADA (*SUPERVISORY CONTROL AND DATA ACQUISITION*) BERBASIS PLC (*PROGRAMMABLE LOGIC CONTROLLER*) UNTUK PENGENDALIAN PINTU AIR**

**SKRIPSI**

**AHMAD FAHLUFI  
0806365444**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
DESEMBER 2010**



**UNIVERSITAS INDONESIA**

**APLIKASI SCADA (*SUPERVISORY CONTROL AND DATA ACQUISITION*) BERBASIS PLC (*PROGRAMMABLE LOGIC CONTROLLER*) UNTUK PENGENDALIAN PINTU AIR**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**AHMAD FAHLUFI  
0806365444**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
DESEMBER 2010**

## HALAMAN PERNYATAAN ORISINALITAS

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

**Nama : AHMAD FAHLUFI**

**NPM : 0806365444**

**Tanda Tangan : **

**Tanggal : 29 Desember 2010**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Ahmad Fahlufi  
N P M : 0806365444  
Program Studi : Teknik Elektro  
Judul Skripsi : Aplikasi SCADA (*Supervisory Control and Data Acquisition*) Berbasis PLC (*Programmable Logic Controller*) Untuk Pengendalian Pintu Air

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

### DEWAN PENGUJI

Pembimbing : Aji Nurwidiyanto ST , MT

(  )

Penguji : Prof. Dr. Ir. Iwa Garniwa M K MT

(  )

Penguji : Ir. Amien Rahardjo MT

(  )

Ditetapkan di : Kampus Universitas Indonesia Depok

Tanggal : 05 Januari 2011

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, saya mengucapkan terima kasih kepada:

- (1) Aji Nur Widyanto S.T., M.T., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi 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, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu

Depok, Desember 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

---

---

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

Nama : Ahmad Fahlufi  
NPM : 0806365444  
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 Noneksklusif** (*Non-exclusive Royalty- Free Right*) atas karya ilmiah saya yang berjudul :

**APLIKASI SCADA (*SUPERVISORY CONTROL AND DATA ACQUISITION*)  
BERBASIS PLC (*PROGRAMMABLE LOGIC CONTROLLER*) UNTUK  
PENGENDALIAN PINTU AIR**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Oktober 2010

Yang menyatakan



( Ahmad Fahlufi )

## ABSTRAK

Nama : Ahmad Fahlufi  
Program Studi : Teknik Elektro  
Judul : Aplikasi SCADA (*Supervisory Control and Data Acquisition*)  
Berbasis PLC (*Programmable Logic Controller*) Untuk  
Pengendalian Pintu Air

Salah satu cara menaggulangi bencana banjir adalah dengan memanfaatkan pintu air sebagai solusinya, tapi beberapa masalah timbul karena pintu air yang ada masih menggunakan cara manual untuk mengamati dan melaporkan kondisi ketinggian air setiap harinya. Cara manual seperti itu akan sangat rentan dengan faktor *Human Error*, dan ditambah lagi dengan faktor alam. Oleh karena itu, dalam skripsi ini akan membahas suatu sistem pemantau dan peringatan dini banjir dengan membuat perancangan sistem SCADA pada sebuah miniatur pintu air yang nantinya bisa membantu mengatasi masalah ini. Pintu air yang menggunakan system SCADA (*Supervisory Control and Data Acquisition*) berbasis PLC (*Programmable Logic Controller*) mampu mengontrol, memonitor serta menyimpan data secara akurat, real time dan efisien. Dilengkapi Alarm antisipasi banjir dan alarm gangguan pintu air serta kamera pengawas. Sensor dan unit beban pada plant yaitu sensor ketinggian air, motor DC, LED, dan buzzer. Perancangan pada HMI (*Human Machine Interface*) SCADA terdapat sistem *Otomatis* dengan mode *auto* dan sistem manual dengan mode *local* dan *remote*. Pada mode *auto* pengaturan beban dilakukan secara *otomatis* berdasarkan dari pembacaan sensor ketinggian air, pintu air akan membuka berurutan berdasarkan ketinggian air per 5 cm dari 0 cm hingga 15 cm, dan kebalikannya. *Sedangkan pada sistem manual* pengontrolan beban dilakukan dari *panel* atau HMI SCADA / *Laptop*. Pengujian sistem dilakukan pengukuran waktu respon dari HMI SCADA terhadap beban. Dari pengujian tersebut disimpulkan bahwa perancangan sistem SCADA pada sebuah miniatur pintu air berhasil dibuat sesuai deskripsi.

### KATA KUNCI :

SCADA, PLC, Pintu air.

## ABSTRACT

Name : Ahmad Fahlufi  
Study Program : Electrical Engineering  
Title : PLC Based Application Of SCADA For Sluice Gate Control

One way of ille-gal flood disaster is to utilize the sluice gates as a solution, but some problems arise because the existing sluice still use manual way to observe and report the condition of the water level every day. How to manuals like it would be very vulnerable to human error factor, and combined with natural factors. Therefore, in this paper will discuss a system of flood monitoring and early warning by making the design of SCADA system on a miniature sluice which later can help overcome this problem. Sluice gates that use SCADA systems (Supervisory Control and Data Acquisition) are based on PLC (Programmable Logic Controller) is able to control, monitor and store data in an accurate, real time and efficient. Equipped Alarms and alarm disturbance anticipated flood water gate and surveillance cameras. Sensors and unit load on the plant that is water level sensors, DC motors, LED and buzzer. The design of the HMI (Human Machine Interface), SCADA systems exist Automatic with auto mode and manual systems with local and remote mode. In auto mode load settings is done automatically based on readings of water level sensor, the floodgates will open a sequence based on the height of water per 5 cm from 0 cm to 15 cm, and the opposite. While the manual system control panel or the load carried from HMI SCADA / Laptop. The test system measurement response time of the HMI SCADA to the load. From the test concluded that the design of SCADA system on a miniature sluice successfully made according to the description.

### KEY WORD :

SCADA, PLC, Sluice Gate



## DAFTAR ISI

HALAMAN JUDUL .....	ii
LEMBAR PERNYATAAN ORISINALITAS .....	iii
LEMBAR PENGESAHAN .....	iv
KATA PENGANTAR .....	v
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS .....	vi
ABSTRAK.....	vii
ABSTRAK.....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
 <b>BAB 1 PENDAHULUAN</b>	
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 .....	3

## **BAB 2 LANDASAN TEORI**

2.1. SCADA ( <i>Supervisory Control and Data Acquisition</i> ) .....	4
2.2. <i>Microsoft Visual Basic 6.0</i> .....	9
2.2.1. <i>Object Timer</i> .....	12
2.2.2. <i>Object Common Dialog</i> .....	13
2.2.3. <i>Tipe Data</i> .....	14
2.2.4. <i>Struktur Keputusan If Then</i> .....	16
2.2.5. <i>Loop For Next</i> .....	16
2.3. <i>Microsoft Access 2007</i> .....	16
2.4. <i>PLC (Programmable Logic Controller)</i> .....	18
2.4.1. <i>PLC Omron CJ1M-CPU11</i> .....	19
2.4.2. <i>PLC Input CJ1W-ID211</i> .....	20
2.4.3. <i>PLC Output CJ1W-OD211</i> .....	20
2.4.4. <i>PLC Power Supply CJ1W-PA202</i> .....	21
2.5. <i>Komunikasi Microsoft Visual Basic 6.0 dengan PLC</i> .....	21
2.6. <i>CX-Programmer 7.0</i> .....	22
2.7. <i>Sensor Ketinggian Air</i> .....	23
2.8. <i>Limit Switch</i> .....	25
2.9. <i>Motor DC</i> .....	25
2.10. <i>Relay DPDT (Double Pole Double Throw)</i> .....	26

## **BAB 3 PERANCANGAN SISTEM**

3.1 <i>Cara Kerja Sistem</i> .....	27
3.2 <i>Perancangan Miniatur Pintu air</i> .....	28
3.3 <i>Perancangan Software</i> .....	37
3.3.1 <i>Perancangan Ladder Diagram</i> .....	37
3.3.2 <i>Perancangan Software HMI (Human Machine Interface)</i> .....	51
3.3.2.1 <i>Form Main Program</i> .....	51
3.3.3 <i>Koneksi Database</i> .....	52

## **BAB 4 PENGUJIAN SISTEM**

4.1 Pengujian Mode Auto.....	53
4.2 Pengujian Mode <i>Local</i> .....	54
4.3 Pengujian Mode <i>Remote</i> .....	56
4.4 Pengujian Alarm.....	58
4.5 Pengujian Database .....	65
4.6 Pengujian Sensor ketinggian Air .....	66

<b>BAB 5 KESIMPULAN.....</b>	<b>67</b>
------------------------------	-----------

<b>DAFTAR REFERENSI .....</b>	<b>68</b>
-------------------------------	-----------

## DAFTAR GAMBAR

2.1	Bagian sistem SCADA sederhana.....	4
2.2	Lingkungan dalam pemrograman <i>Visual Basic 6.0</i> .....	9
2.3	Toolbar .....	10
2.4	Toolbox.....	10
2.5	<i>Form Window</i> .....	11
2.6	<i>Project Explorer</i> .....	11
2.7	<i>Properties Window</i> .....	12
2.8	<i>Object Timer</i> .....	13
2.9	<i>Properties Object Timer</i> .....	13
2.10	<i>Object Common Dialog</i> .....	14
2.11	CJ1M-CPU11 .....	20
2.12	PLC <i>Input</i> CJ1W-ID211 .....	20
2.13	PLC <i>Output</i> CJ1W-OD211 .....	21
2.14	PLC <i>Power Supply</i> CJ1W-PA202.....	21
2.15	Kontak NO.....	22
2.16	Kontak NC.....	22
2.17	<i>Coil</i> .....	22
2.18	<i>Timer</i> .....	23
2.19	<i>MOV</i> .....	23
2.20	Cara Kerja Sensor Ketinggian Air .....	24
2.21	Diagram Kerja Sensor Ketinggian Air .....	24
2.22	<i>Limit Switch</i> .....	25
2.23	Motor D.C Sederhana .....	26
2.24	Model <i>Relay</i> .....	26
3.1	Blok diagram cara kerja sistem .....	27
3.2	Desain Kerangka.....	28
3.3	Desain Pintu.....	29
3.4	Indikator Ketinggian Air.....	30
3.5	<i>Box Panel</i> .....	31
3.6	Plant Keseluruhan .....	31
3.7	Skema Cara Kerja Sensor Air .....	32

3.8	Rangkaian Sensor Ketinggian Air .....	33
3.9	Skematik rangkaian gambar <i>relay motor</i> .....	34
3.10	Skematik Rangkaian Gambar <i>Relay Sensor</i> .....	34
3.11	Modul Rangkaian <i>Relay</i> .....	35
3.12	<i>Geer box</i> .....	35
3.13	<i>Motor DC</i> .....	36
3.14	Sensor Pintu .....	36
3.15	<i>Flowchart</i> Untuk Sistem Otomatis.....	39
3.16	<i>Ladder Mode Otomatis</i> .....	40
3.17	<i>Ladder Mode Otomatis (Lanjutan)</i> .....	41
3.18	<i>Flowchart Local System</i> Proses Buka .....	42
3.19	<i>Flowchart Local System</i> Proses Tutup .....	43
3.20	<i>Flowchart Remote System</i> Buka.....	44
3.21	<i>Flowchart Remote System</i> Tutup .....	45
3.22	<i>Ladder Mode Local</i> .....	46
3.23	<i>Ladder Mode remote</i> .....	47
3.24	<i>Flowchart Alarm</i> Antisipasi Banjir .....	48
3.25	<i>Flowchart Alarm</i> Gangguan Pintu Air .....	49
3.26	<i>Ladder Alarm</i> .....	50
3.27	<i>Contact alarm</i> dari visual basic.....	51
3.28	<i>Form Main Program</i> .....	52
3.29	<i>ADODC</i> .....	52
4.1	Tampilan Alarm Hidup Dengan Pembacaan Sensor Ketinggian Air .....	59
4.2	Alarm Hidup Dengan Menaikan Ketinggian Air Secara <i>Manual</i> .....	60
4.3	Tampilan Alarm Hidup Karena Ada Kerusakan Pada Pintu 1 .....	61
4.4	Tampilan Tanda Peringatan Jika Kerusakan Pada Salah Satu Pintu.....	62
4.5	Database Untuk Kondisi Plant .....	63
4.6	Database Ketinggian Air Pada Plant .....	64
4.7	Database Ketinggian Air Pada Microsoft Excel .....	64

## DAFTAR TABEL

2.1 Pilihan pada <i>common dialog</i> .....	14
2.2 Tipe Data.....	15
3.1 <i>Input</i> PLC CJ1W-ID211 yang digunakan.....	37
3.2 <i>Output</i> PLC CJ1W-OD211 yang digunakan.....	38
4.1 Hasil Pengujian 1 Program Deteksi Ketinggian Air .....	53
4.2 Hasil Pengujian 1 Respon Dari Tombol Panel.....	55
4.3 Hasil Pengujian 2 Kecepatan Buka Tutup Pintu pada mode <i>local</i> .....	55
4.4 Hasil Pengujian 1 Respon Dari Tombol HMI SCADA .....	56
4.5 Hasil Pengujian 2 Kecepatan Buka Tutup Pintu pada mode <i>remote</i> .....	57
4.6 Hasil Pengujian Respon Tombol Setting Water HMI SCADA .....	57
4.7 Hasil Pengujian Respon Program <i>CCTV</i> .....	58
4.8 Hasil Pengujian Sensor Ketinggian Air Cara 1 .....	65
4.9 Hasil Pengujian Sensor Ketinggian Air Cara 2.....	66

## 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

Banjir adalah bencana yang sering terjadi salah satu cara menanggulangi bencana banjir adalah dengan memanfaatkan pintu air sebagai solusinya, tapi beberapa masalah timbul karena pintu air yang ada masih menggunakan cara manual untuk mengamati dan melaporkan kondisi ketinggian air setiap harinya. Cara manual seperti itu akan sangat rentan dengan faktor *Human Error*, dan ditambah lagi dengan faktor alam. Oleh karena itu dalam tugas akhir ini akan dibuat perancangan sistem SCADA (*Supervisory Control and Data Acquisition*) untuk mengawasi dan mengontrol unit-unit beban pada miniatur pintu air yaitu berupa pengontrolan *motor* penggerak pintu air berdasarkan ketinggian air pada tabung yang dideteksi oleh sensor ketinggian air.

Penerapan sistem SCADA ini dimaksudkan untuk mengawasi dan mengontrol penggunaan unit-unit beban pada miniatur pintu air agar dapat dilakukan efisiensi dalam pengontrolan serta memonitor unit-unit beban tersebut. Dalam sistem ini akan dilakukan akuisisi data yaitu untuk aktifitas pada pembacaan ketinggian air pada tabung indikator air yang terdapat pada miniatur pintu air banjir kanal. Data tersebut diolah untuk mengetahui ketinggian air serta kondisi pintu air pada *plant* miniatur pintu air banjir kanal. 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 miniature pintu air. 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* induk dan *Microsoft Excel 2007* sebagai penyimpan *database* ke-dua.

## 1.3 Pembatasan Masalah

Adapun hal-hal yang membatasi tugas akhir ini adalah:

1. Unit-unit beban pada miniatur pintu air yang diawasi dan dikontrol secara *real time* hanya pintu air dan kamera pengawas.
2. Pintu air yang diawasi dan dikontrol hanya beberapa saja sebagai simulasi pada miniatur pintu air.
3. Sensor yang diawasi dan dikontrol yaitu sensor pendeteksi arah putaran untuk medeteksi tinggi air dan limit switch sebagai pendeteksi kondisi pintu air. Pembahasan tentang sensor atau rangkaian pembaca sensor tidak dibahas secara lengkap.
4. Sistem akuisisi datanya hanya berupa aktivitas dari pengoperasian pintu air dan hasil pendeteksian ketinggian air pada miniatur pintu air.
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 pintu air
2. Untuk mendapatkan model perancangan pintu air yang canggih dan modern

## 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*. Selanjutnya bab 3 perancangan sistem membahas tentang perancangan sistem mulai dari desain miniatur pintu air, 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

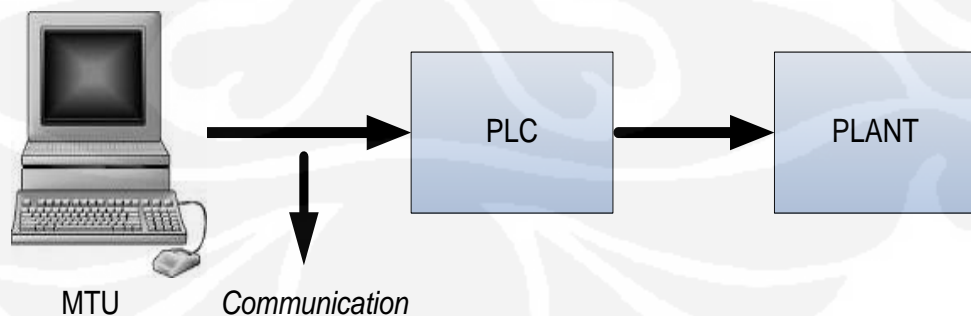
### LANDASAN TEORI

#### 2.1 SCADA <sup>[1]</sup>

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.

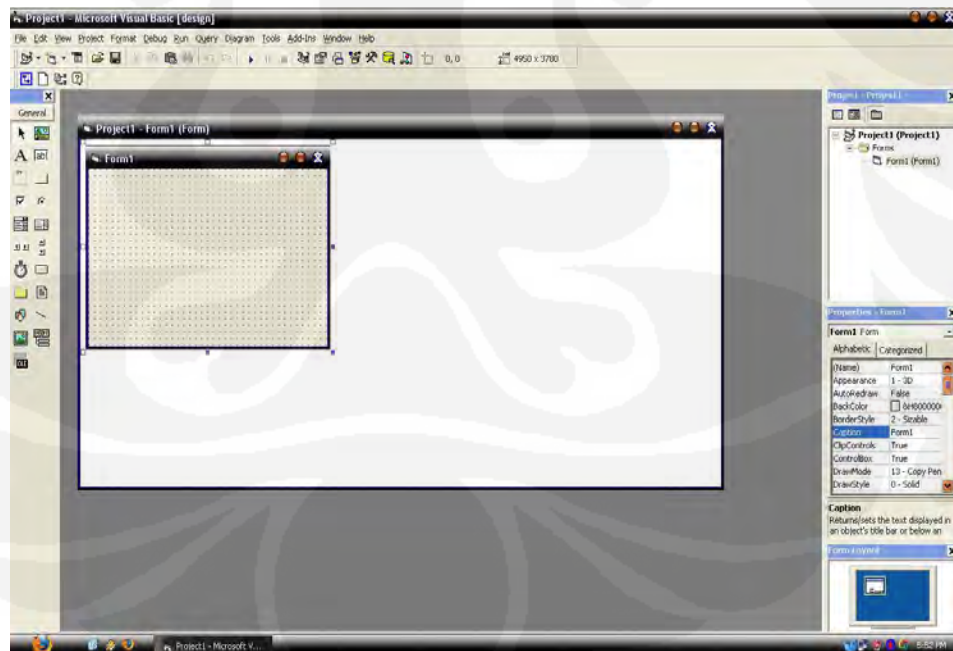
Untuk akuisisi data pada SCADA, data yang didapat berasal dari sensor-sensor yang terdapat di *plant*. Pada sistem SCADA yang kompleks mungkin dibutuhkan pemantauan terhadap ratusan hingga ribuan sensor yang tersebar di seluruh area yang terdiri dari beberapa *plant*.

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 <sup>[4]</sup>

*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*

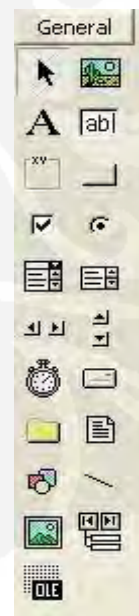
*Toolbar* merupakan sebuah komponen yang berisi kumpulan tombol yang terletak dibagian bawah menu bar. Pada kondisi default program *Visual Basic* hanya menampilkan toolbar standar.



Gambar 2.3 *Toolbar*

### 2. *Toolbox*

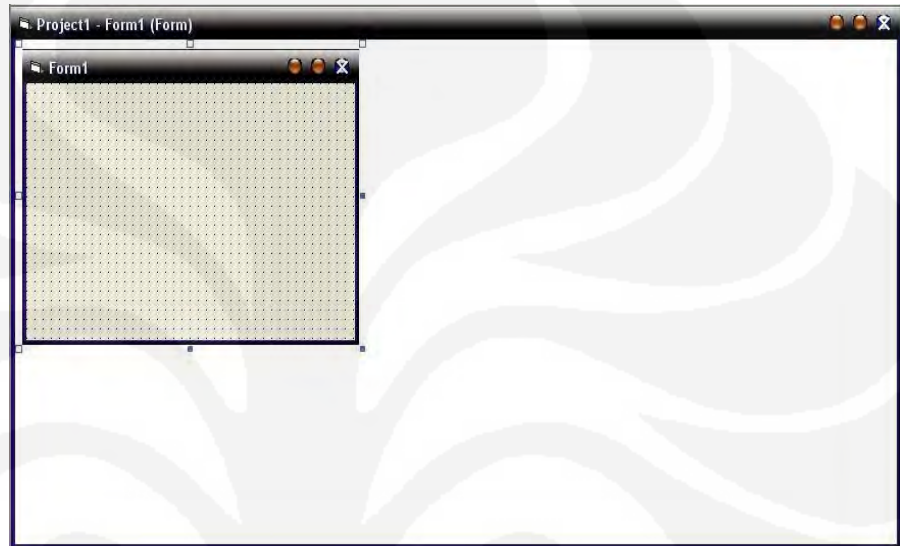
*Toolbox* adalah suatu objek yang terdiri dari elemen-elemen antarmuka program untuk ditambahkan ke dalam form. *Toolbox* biasanya terletak di sebelah kiri layar dan mengandung kontrol-kontrol yang bisa digunakan untuk menambah gambar, label, tombol, kotak daftar, menu serta bentuk-bentuk geometris pada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



Gambar 2.4 *Toolbox*

### 3. *Form Window*

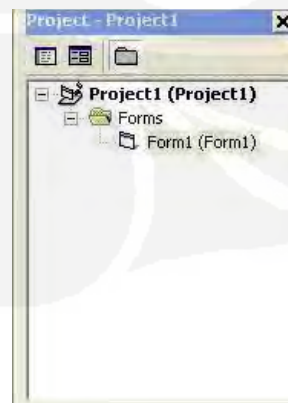
*Form Window* adalah daerah kerja utama, dimana dapat digunakan untuk membuat program-program aplikasi *Visual Basic*. Pada form ini dapat ditempatkan berbagai macam objek misalnya teks, gambar, tombol-tombol perintah, *database*, *combo box* dan lain-lain.



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.



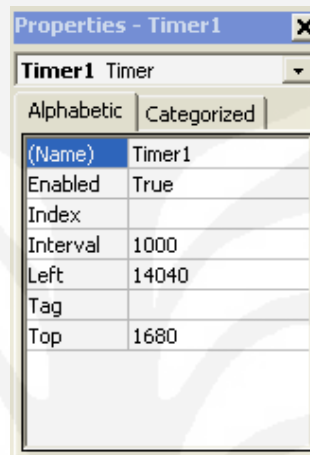
Gambar 2.7 Properties Window

### 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*

<b>Kotak Dialog</b>	<b>Tujuan</b>	<b>Metode</b>
<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

<b>Tipe Data</b>	<b>Ukuran</b>	<b>Kisaran</b>
<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<sup>[6]</sup>

*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<sup>[3]</sup>

*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<sup>[10]</sup>

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<sup>[4]</sup>

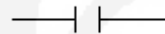
PLC dapat berkomunikasi dengan komputer dengan menggunakan *Host Link*. *Host Link* ini merupakan *interface* PLC terhadap *host* komputer. Melalui *Host Link* seluruh area memori dalam PLC dapat diakses termasuk memori program. Dalam tugas akhir ini digunakan *Host link* RS-232 yang dapat dihubungkan 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<sup>[10]</sup>

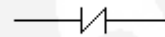
Program ini digunakan untuk membuat *ladder diagram* pada PLC Omron CJ1M. dibawah ini terdapat beberapa instruksi yang ada pada CX-Programmer 7 :

### 1. Kontak NO (*Normaly Open*)



Gambar 2.15 Kontak NO

### 2. Kontak NC (*Normaly Close*)



Gambar 2.16 Kontak NC

### 3. *Coil*

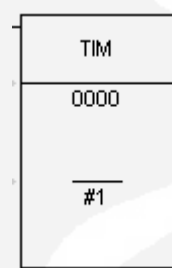
Digunakan sebaail *coil relay* bantu dan juga *output*.



Gambar 2.17 *Coil*

#### 4. *Timer*

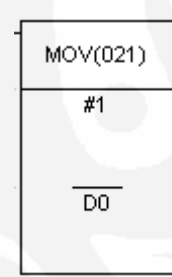
Digunakan untuk mengatur waktu *On-Off* dari kontak *timer* tersebut, sesuai dengan waktu *setting* dari *timer* tersebut. #1 pada gambar dimaksud waktu *setting timer* adalah 0.1 s



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).



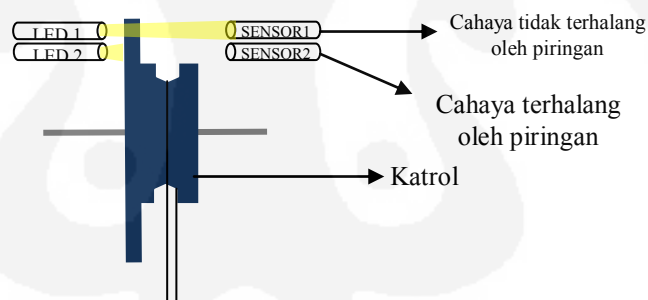
Gambar 2.19 *MOV*

### 2.7 Sensor Ketinggian Air <sup>[5]</sup>

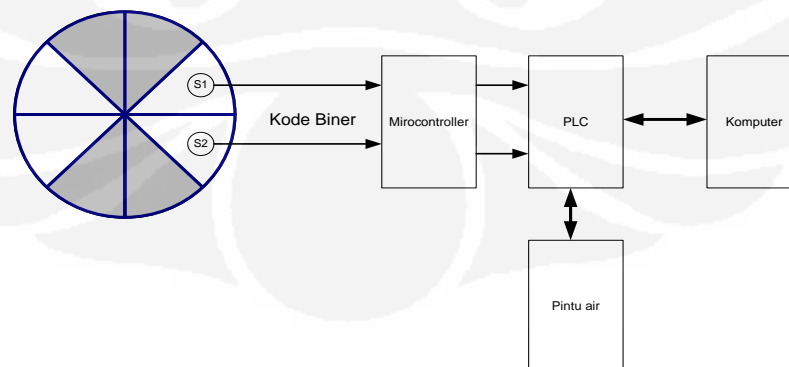
Sensor ketinggian air ini adalah sensor yang berfungsi untuk mendeteksi ketinggian air, sensor ketinggian air ini bekerja secara *digital* cara kerja sensor ini terletak pada dua buah sensor cahaya yang di halangi piringan dan di bantu dengan mikrokontroler dalam pembacaan kode binernya, Piringan pada sensor air berbentuk lingkaran yang di bagi menjadi delapan bagian, empat bagian tembus cahaya atau bening sedangkan empat bagian lainnya tidak tembus cahaya. Pada piringan terdapat dua led sebagai pengirim cahaya

dan terdapat juga dua sensor optic cahaya yang akan mengeluarkan kode biner sehingga dapat memberikan perbedaan dari setiap pergerakan putaran piringan yang bergerak berdasarkan naik turunnya bandul. Jika air naik maka bandul dalam pipa akan naik dan bandul luar akan turun, tali yang dikatkan pada bandul dililitkan pada piringan sensor air sehingga jika bandul naik maka piringan sensor air akan memutar searah jarum jam dan sebaliknya jika air turun, sensor tersebut memberikan sinyal kode biner ke *microcontroller*, dan *microkontroller* akan membaca kode biner dari *output* sensor.

Kode biner yang diterima *microcontroller* akan di proses untuk memudahkan pembacaan kode pada PLC , PLC akan memberikan data ke komputer dan melakukan control ke Plant pintu, Pada plant pintu terdapat beberapa sensor yang akan mendeteksi pergerakan pintu membatasi pergerakan pintu sehinga pengontrolan PLC dapat berjalan dengan baik. Berikut adalah gambar cara kerja dari sensor ketinggian air serta diagram kerja dari sensor ketinggian air:



Gambar 2.20 Cara Kerja Sensor Ketinggian Air



Gambar 2.21 Diagram Kerja Sensor Ketinggian Air

## 2.8. *Limit Switch*<sup>[9]</sup>

*Limit switch* adalah suatu tombol atau katup atau indikator mekanik yang diletakkan pada suatu tempat yang digerakkan ketika suatu bagian mekanik berada di ujung sesuai dengan pergerakan yang diinginkan. *Limit switch* ini banyak digunakan dalam industri sebagai sensor pembatas yang akan membatasi pergerakan dari suatu mesin sampai batas yang telah ditetapkan. Di bawah ini adalah salah satu contoh dari *Limit switch* :



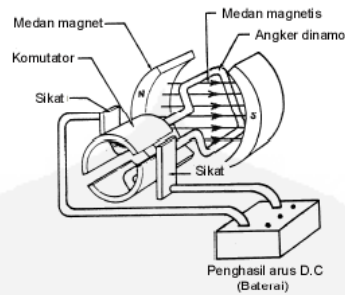
Gambar 2.22 *Limit Switch*

## 2.9. *Motor DC*<sup>[8]</sup>

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya memutar *impeller* pompa, *fan* atau *blower*, dll.

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bias berputar bebas di antara kutub-kutub magnet permanen.





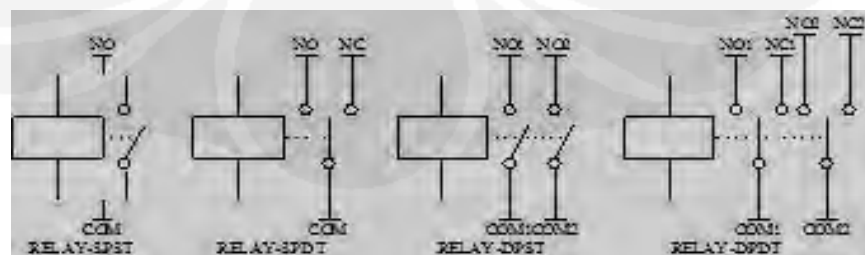
Gambar 2.23 Motor D.C Sederhana

### 2.10. Relay DPDT (Double Pole Double Throw) <sup>[7]</sup>

Relay adalah saklar yang dikendalikan secara elektronik (*electronically switch*). Arus listrik yang mengalir pada kumparan relay akan menciptakan medan magnet yang kemudian akan menarik lengan relay dan mengubah posisi saklar, yang sebelumnya terbuka menjadi terhubung. Relay memiliki tiga jenis kutub: COMMON = kutub acuan, NC (*Normally Close*) = kutub yang dalam keadaan awal terhubung pada COMMON, dan NO (*Normally Open*) = kutub yang pada awalnya terbuka dan akan terhubung dengan COMMON saat kumparan relay diberi arus listrik. Berdasarkan jumlah kutub pada relay, maka relay dibedakan menjadi 4 jenis:

- SPST = *Single Pole Single Throw*
- SPDT = *Single Pole Double Throw*
- DPST = *Double Pole Single Throw*
- DPDT = *Double Pole Double Throw*

*Pole* adalah jumlah COMMON, sedangkan *Throw* adalah jumlah terminal output (NO dan NC).



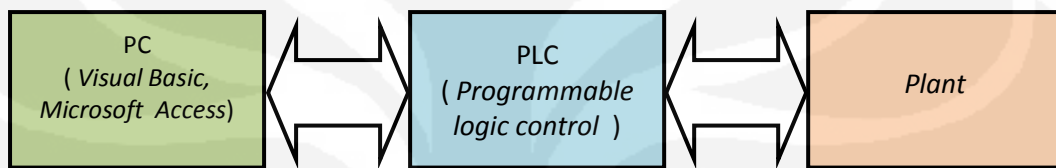
Gambar 2.24 Model Relay

## 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

Gambar di atas merupakan aplikasi dari SCADA sederhana yang menggunakan satu PLC untuk mengendalikan dan memonitoring *plant*. Dalam diagram tersebut digambarkan PLC dapat dikontrol dengan komputer dengan menggunakan HMI (*Human Machine Interface*), HMI ini dibuat dengan bahasa pemrograman *Visual Basic 6.0*. Kemudian untuk pengolahan databasenya digunakan *Microsoft Access 2007* dan *Microsoft Excel 2007*.

Pada bagian *plant* terdapat beberapa beban dan sensor, beban yang harus di kontrol pada miniatur *plant* banjir kanal ini yaitu berupa beberapa *relay* yang berfungsi untuk mengaktifkan *motor* serta mengatur arah putaran *motor* sehingga pintu air dapat membuka atau menutup.

Sedangkan sensor terdiri dari beberapa sensor, sensor yang pertama adalah sensor untuk mengetahui tinggi dari air pada tabung pengukur, dan yang kedua adalah sensor untuk memberhentikan *motor* jika pintu menutup atau membuka.

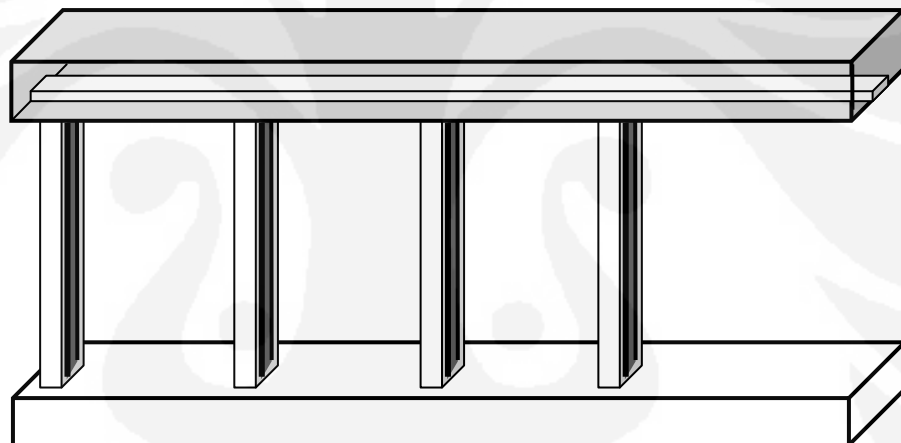
### 3.2 Perancangan Miniatur Pintu Air

Pada perancangan ini akan dibuat sebuah miniatur pintu air dengan menggunakan bahan *acrylic*, Alumunium, dan beberapa bahan pendukung .

Desain Minatur:

#### 1) Kerangka:

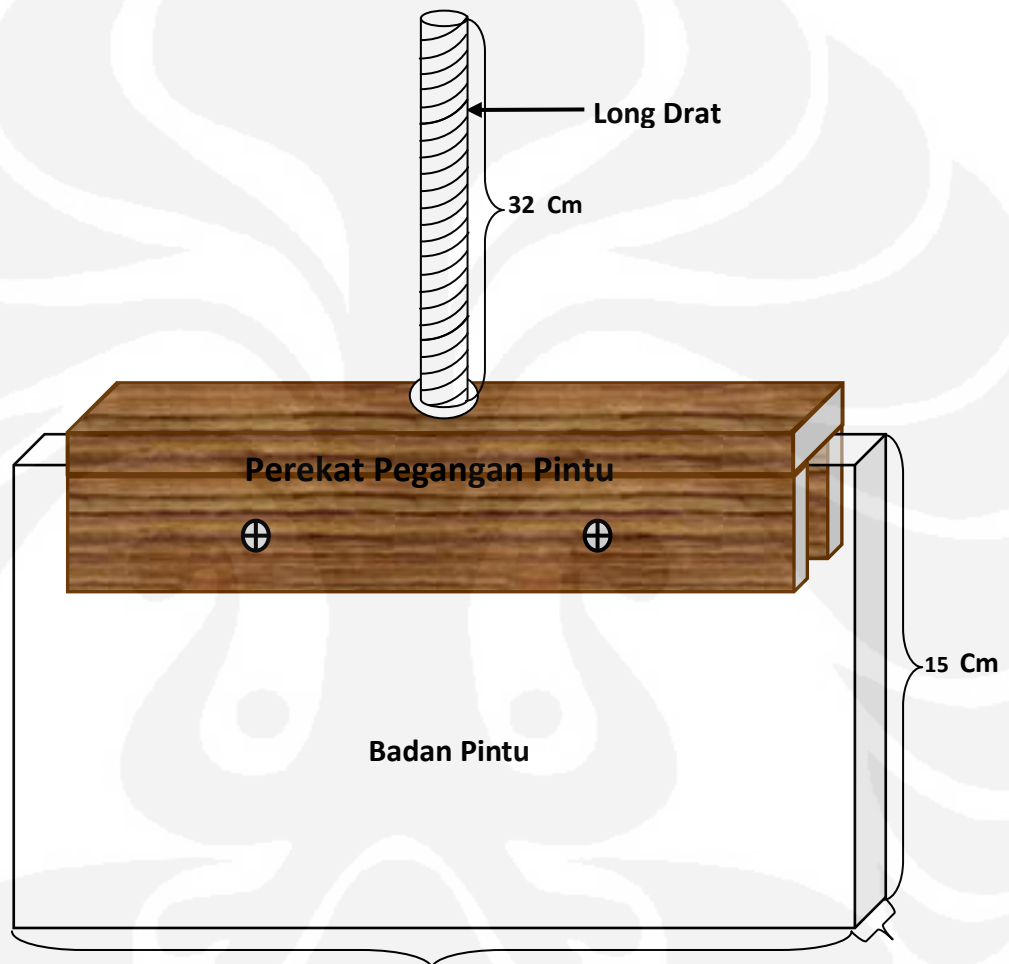
Kerangka pada plant menggunakan bahan alumunium dengan ketebalan 1 mm yang biasa digunakan pada pembuatan kusain untuk jendela alumunium serta memakai alumunium yang biasa digunakan untuk pengait gorden. Dengan pemakaian bahan-bahan tersebut maka dapat memudahkan dalam penempatan perangkat-perangkat lain yang akan digunakan.



Gambar 3.2 Desain Kerangka

## 2) Pintu:

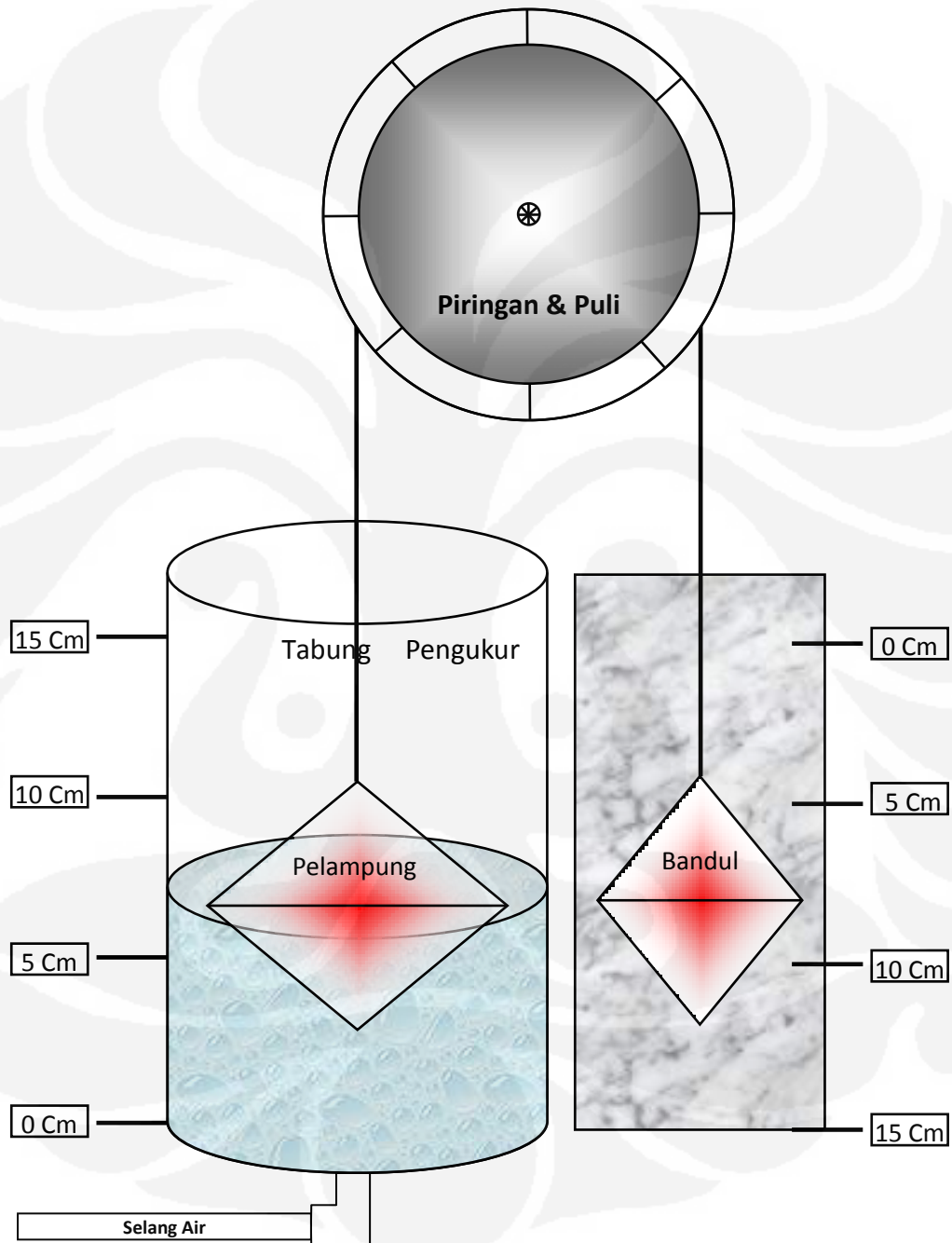
Pintu menggunakan bahan acrylic bening dengan ketebalan 5 mm, dengan long drat sebagai ngait gerak dan kayu yang sebagai perekat antara acrylic dengan long drat.



Gambar 3.3 Desain Pintu

### 3) Indikator Ketinggian Air:

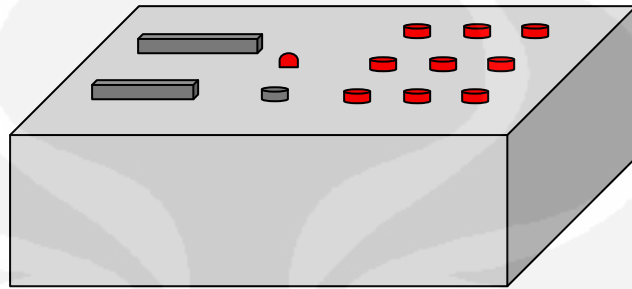
Indikator ketinggian air terdiri dari piringan plastik, tabung plastik untuk penampungan air, dan papan *acrylic* sebagai tabel baca ketinggian air.



Gambar 3.4 Indikator Ketinggian Air

#### 4) *Box Panel*

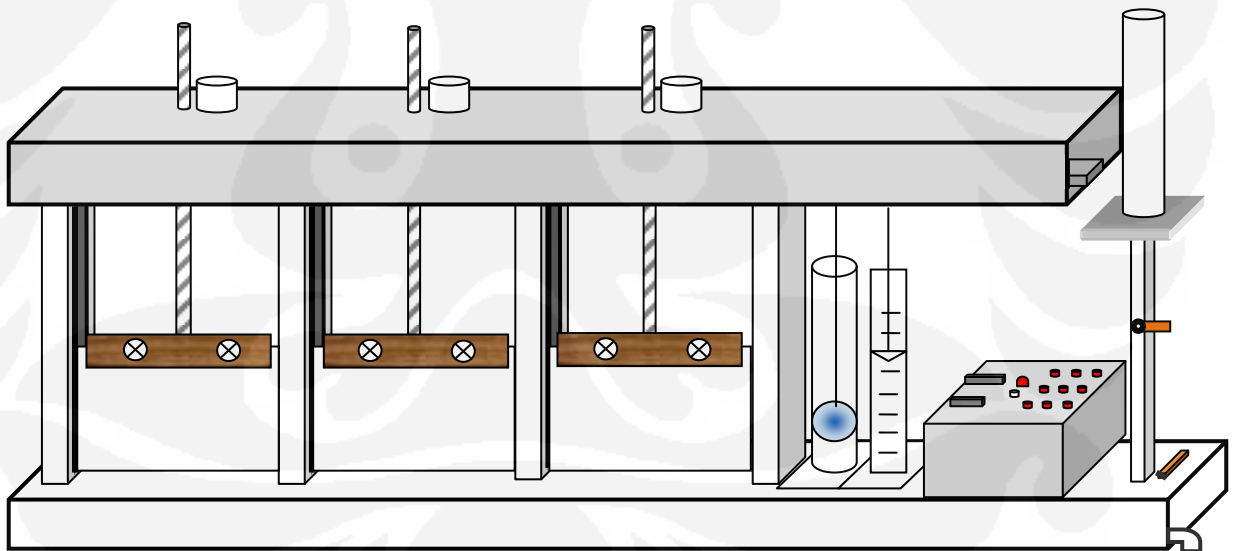
*Box Panel* digunakan sebagai tempat menyimpan rangkaian sekaligus digunakan sebagai *panel* untuk kontrol manual dan sebagai terminal dari PLC dan catu daya.



Gambar 3.5 *Box Panel*

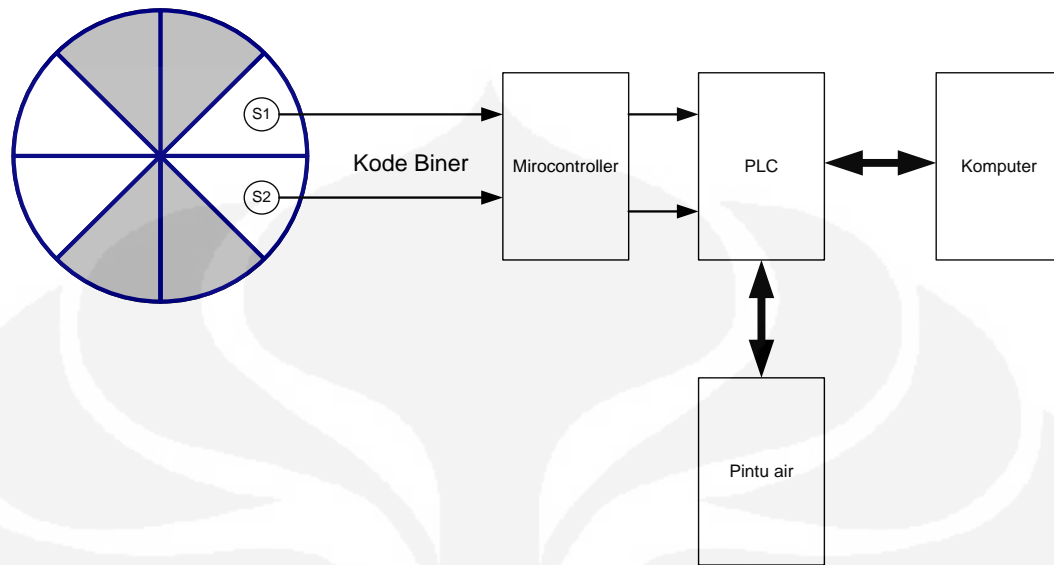
#### 5) Plant Keseluruhan

Berikut ini adalah gambar plant secara keseluruhan .



Gambar 3.6 Plant Keseluruhan

## 6) Rangkaian Sensor Ketinggian Air:



Gambar 3.7 Skema Cara Kerja Sensor Air

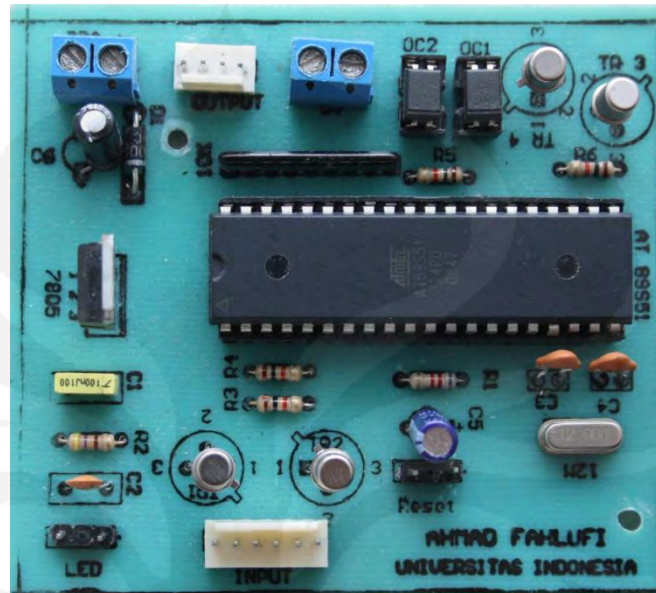
Cara kerja Piringan sensor air berbentuk lingkaran yang di bagi menjadi delapan bagian, empat bagian tembus cahaya atau bening sedangkan empat bagian lainnya tidak tembus cahaya.

Pada piringan terdapat dua led sebagai pengirim cahaya dan terdapat juga dua sensor *optic* cahaya yang akan mengeluarkan kode biner sehingga dapat memberikan perbedaan dari setiap pergerakan putaran piringan yang bergerak berdasarkan naik turunnya bandul.

Jika air naik maka bandul dalam pipa akan naik dan bandul luar akan turun, tali yang dikatkan pada bandul dililitkan pada piringan sensor air sehingga jika bandul naik maka piringan sensor air akan memutar searah jarum jam dan sebaliknya jika air turun, sensor tersebut memberikan sinyal kode biner ke *microcontroller*, dan mikrokontroller akan membaca kode biner dari *output* sensor.

Kode biner yang diterima *microcontroller* akan di proses untuk memudahkan pembacaan kode pada PLC , PLC akan memberikan data ke komputer dan melakukan control ke Plant pintu, Pada plant pintu terdapat beberapa sensor yang akan mendeteksi pergerakan pintu membatasi pergerakan pintu sehingga pengontrolan PLC dapat berjalan dengan baik.

Gambar di bawah ini merupakan rangkaian *driver* dari sensor ketinggian air.

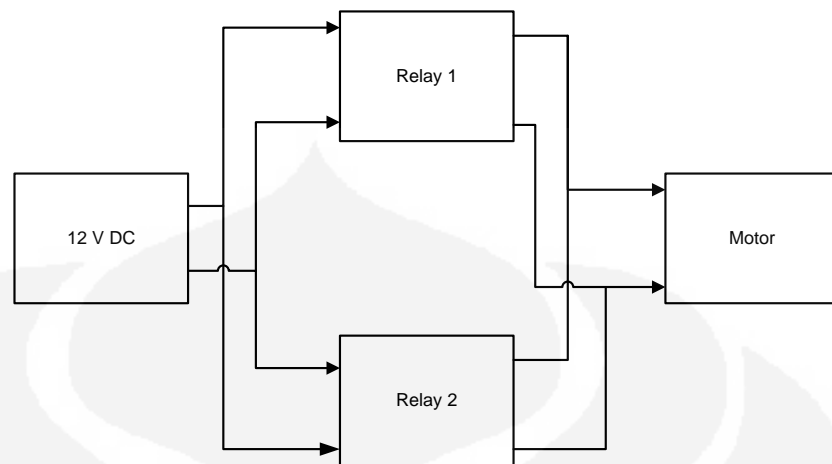


Gambar 3.8 Rangkaian Sensor Ketinggian Air

#### 7) Rangkaian *Relay*

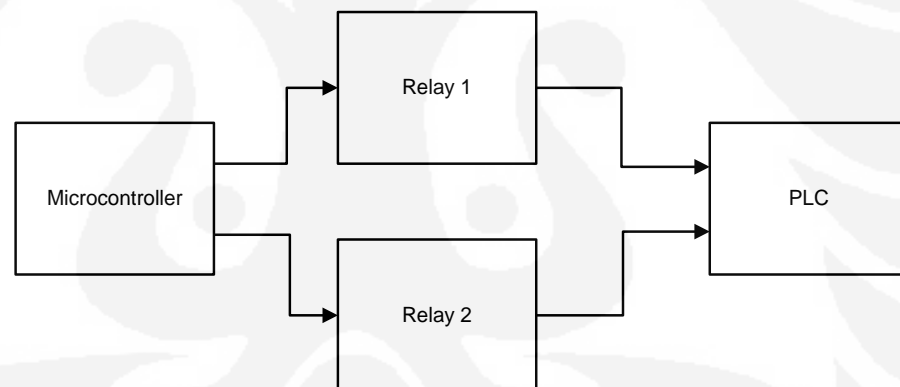
Rangkaian *Relay* ini terdapat delapan *relay* yang dibagi menjadi dua fungsi. Fungsi yang pertama yaitu untuk mengaktifkan *motor* sesuai dengan kendali dari PLC pada bagian ini terdapat enam buah *relay* yang digunakan untuk mengendalikan tiga buah *motor*, pada setiap *motor* menggunakan dua buah *relay* dengan tipe DPDT (*Double Pole Double Throw*) yang dapat menyambungkan dan memutus dua sumber. Sedangkan fungsi yang ke dua yaitu sebagai pemberi *input* ke PLC dari rangkaian sensor ketinggian air, di bagian ini terdapat dua buah *relay* yang di kendalian oleh rangkaian sensor ketinggian air. Berikut adalah skematik rangkaian gambar dari *relay* yang digunakan pada *motor*:





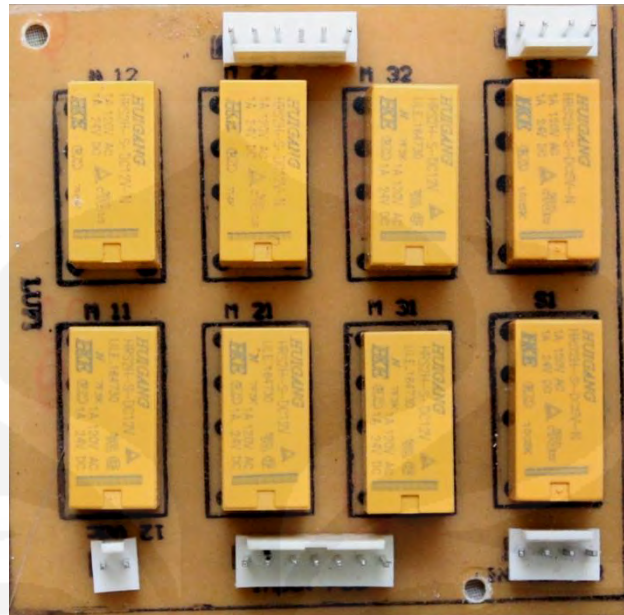
Gambar 3.9 Skematik rangkaian gambar *relay motor*

Berikut adalah skematik rangkaian gambar dari *relay* yang digunakan pada sensor untuk memberikan *input* ke PLC:



Gambar 3.10 Skematik Rangkaian Gambar *Relay Sensor*

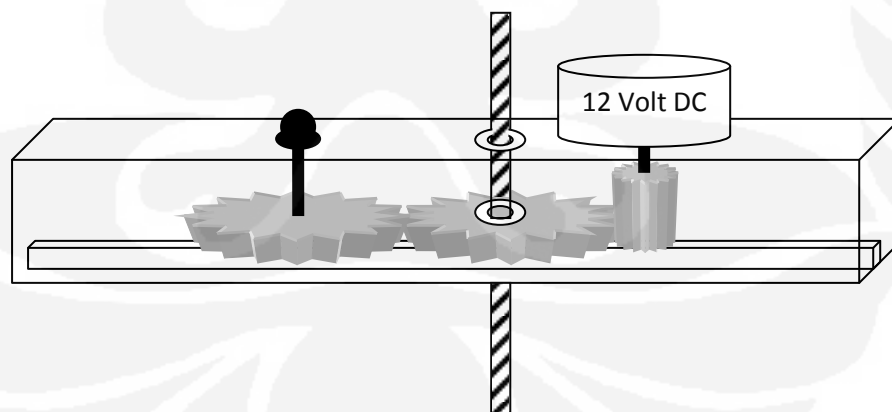
Berikut adalah rangkaian gambar dari *relay*:



Gambar 2.11 Modul Rangkaian *Relay*

#### 8) *Gear Box* dan *Motor*

*Gear box* ini digunakan untuk memudahkan *motor* dalam melakukan proses pengangkatan atau penurunan pintu air. *Motor* yang digunakan adalah *motor* 12 Volt DC. *Gear* ini diletakan di dalam kerangka atas *plant* miniatur banjir kanal.



Gambar 2.12 *Geer box*

Sedangkan *Motor* yang digunakan untuk penggerak pintu adalah *motor* DC 12 Volt, yang biasa digunakan untuk penggerak pada radio

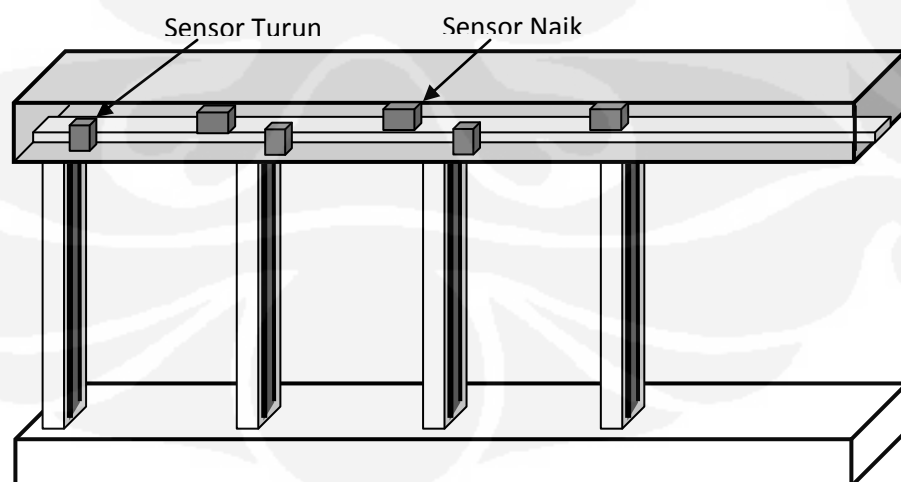
*tape*. Untuk dapat menggerakkan pintu maka pada as ujung *motor* digunakan *gear* dengan ukuran diameter 3 cm. *Gear* ini akan berkaitan dengan *gear* pada as pintu yang berbentuk ulir dengan perbandingan *gear* pada *ash* pintu dan pada *ash motor* yaitu 1:5. Perbandingan ini dipakai agar pergerakan naik maupun turunnya pintu menjadi lebih baik. Berikut adalah salah satu gambar dari *motor dc*:



Gambar 2.13 Motor DC

#### 9) Sensor Pintu

Sensor yang di gunakan pada pintu yaitu *limit switch*, yang dimodifikasi sehingga dapat mendeteksi keadaan pintu ketika pintu telah membuka maksimal atau menutup. *Limit switch* yang di gunaka berjumlah enam buah dengan dua fungsi berbeda, tiga *buah limit switch* di gunakan untuk mendeteksi pintu naik, dan tiga *limit switch* lagi di gunakan untuk mendeteksi pintu turun.



Gambar 3.14 Sensor Pintu

### 3.3 Perancangan *Software*

Perancangan software ini terdiri dari dua bagian yaitu perancangan *ladder diagram* untuk menjalankan PLC dan perancangan HMI. Perancangan *software* ini saling terkait satu sama lain, karena tanpa perancangan *ladder diagram* terlebih dahulu nanti akan mengalami kesulitan dalam membuat *software* HMI-nya. Perancangan *ladder diagram* ini menggunakan *software* CX-Programmer versi 7 keluaran Omron. Kemudian untuk membuat *software* HMI-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. Kemudian untuk penyimpanan databasenya digunakan *Microsoft Access* 2007.

#### 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. Berikut ini adalah tabel I/O PLC yang dipakai :

Tabel 3.1 *Input* PLC CJ1W-ID211 yang digunakan

TABEL *INPUT* PLC

No	<i>Input</i>	Keterangan
1	0.00	<i>Relay</i> Sensor Air Naik
2	0.01	<i>Relay</i> Sensor Air Turun
3	0.02	Sensor Pintu 1 Naik
4	0.03	Sensor Pintu 1 Turun
5	0.04	Sensor Pintu 2 Naik
6	0.05	Sensor Pintu 2 Turun
7	0.06	Sensor Pintu 3 Naik
8	0.07	Sensor Pintu 3 Turun
9	0.08	<i>Push Bottom</i> Pintu 1 Buka Manual

10	0.09	<i>Push Bottom Pintu 1 Tutup Manual</i>
11	0.10	<i>Push Bottom Pintu 2 Buka Manual</i>
12	0.11	<i>Push Bottom Pintu 2 Tutup Manual</i>
13	0.12	<i>Push Bottom Pintu 3 Buka Manual</i>
14	0.13	<i>Push Bottom Pintu 3Tutup Manual</i>
15	0.14	<i>Push Bottom Alarm Manual</i>
16	0.15	<i>Push Bottom Reset Manual</i>

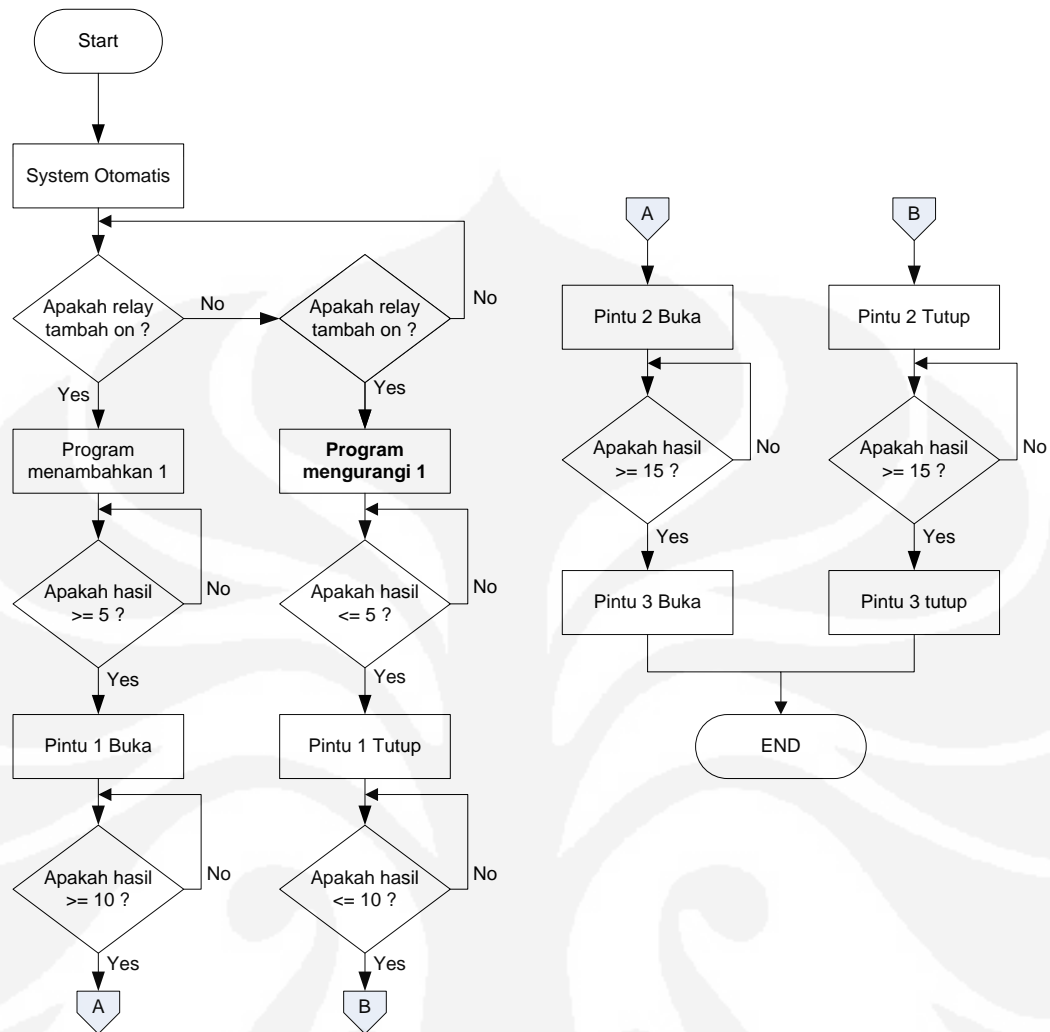
Tabel 3.2 *Output* PLC CJ1W-OD211 yang digunakanTABEL *OUTPUT* PLC

No	<i>Output</i>	Keterangan
1	1.00	Pintu 1 Naik
2	1.01	Pintu 1 Turun
3	1.02	Pintu 2 Naik
4	1.03	Pintu 2 Turun
5	1.04	Pintu 3 Naik
6	1.05	Pintu 3 Turun
7	1.06	<i>Alarm ( LED dan Buzzer)</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 tiga mode yaitu *Auto*, *local* dan *remote*. Berikut ini adalah *flowchart* cara kerja kedua sistem tersebut.

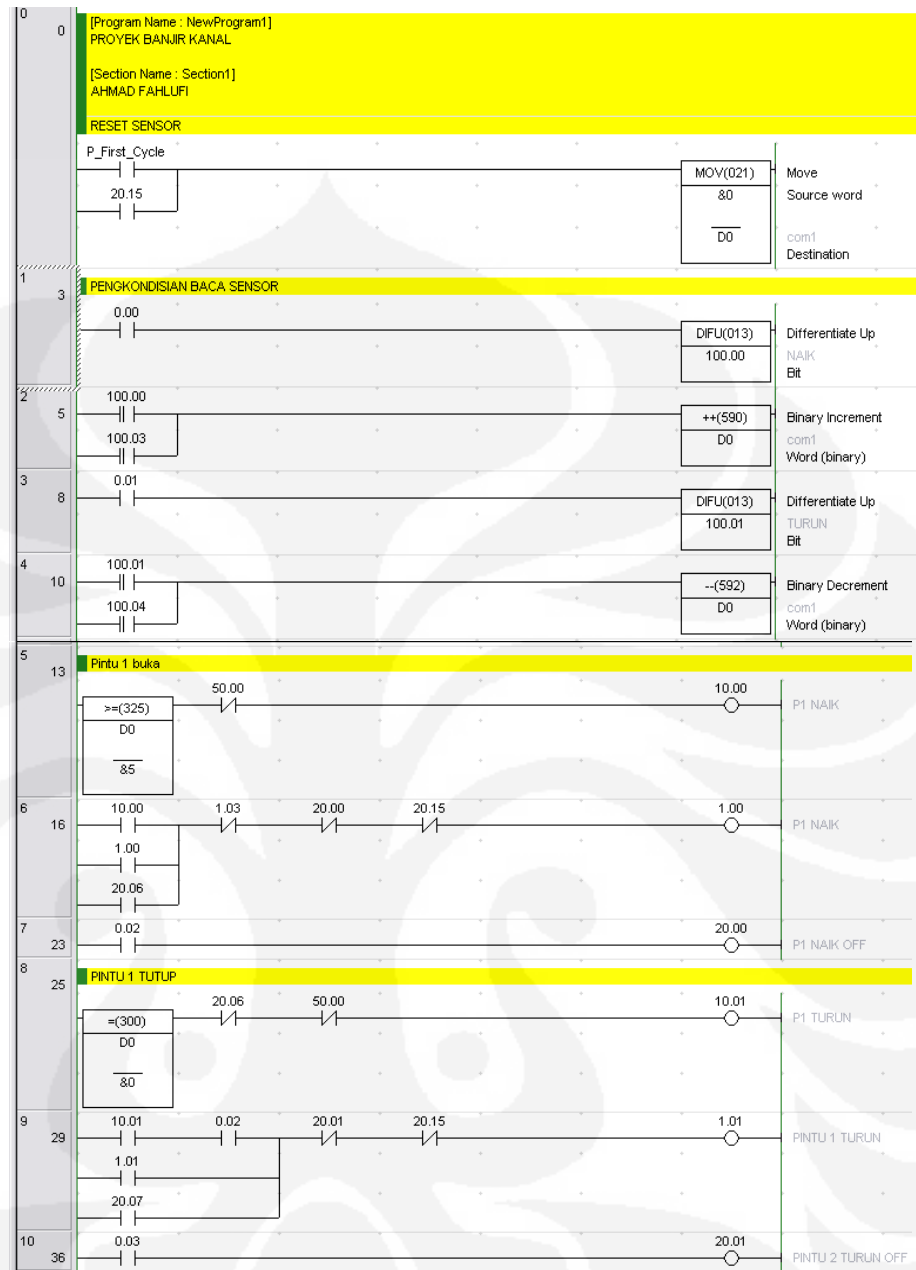
➤ **Mode *Auto***

Pada Mode *Auto* sistem dapat memungkinkan buka tutup pintu secara otomatis berdasarkan sensor ketinggian air. Di plant ini juga terdapat Alarm yang berfungsi sebagai pendeteksi banjir karena alarm akan berbunyi jika ketinggian air melampaui batas normal. Berikut ini *flowchart* untuk cara kerja sistem tersebut:

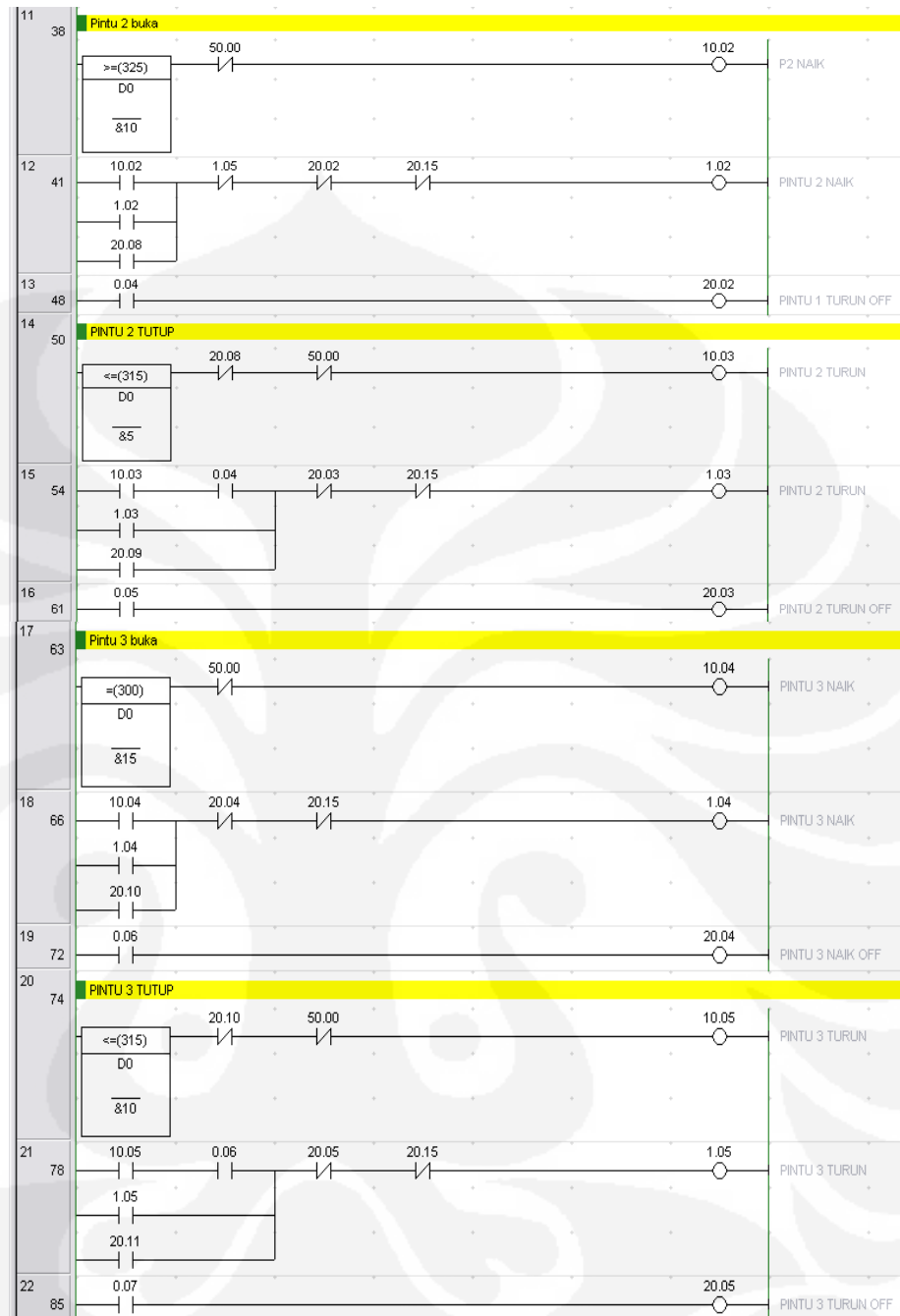


Gambar 3.15 *Flowchart* untuk sistem Otomatis

Setelah merancang diagram alir mode Otomatis maka langkah selanjutnya adalah mengimplementasikan ke dalam bentuk *Ladder mode Otomatis*. Berikut adalah *ladder*-nya:



Gambar 3.16 Ladder Mode Otomatis



Gambar 3.17 Ladder Mode Otomatis (Lanjutan)

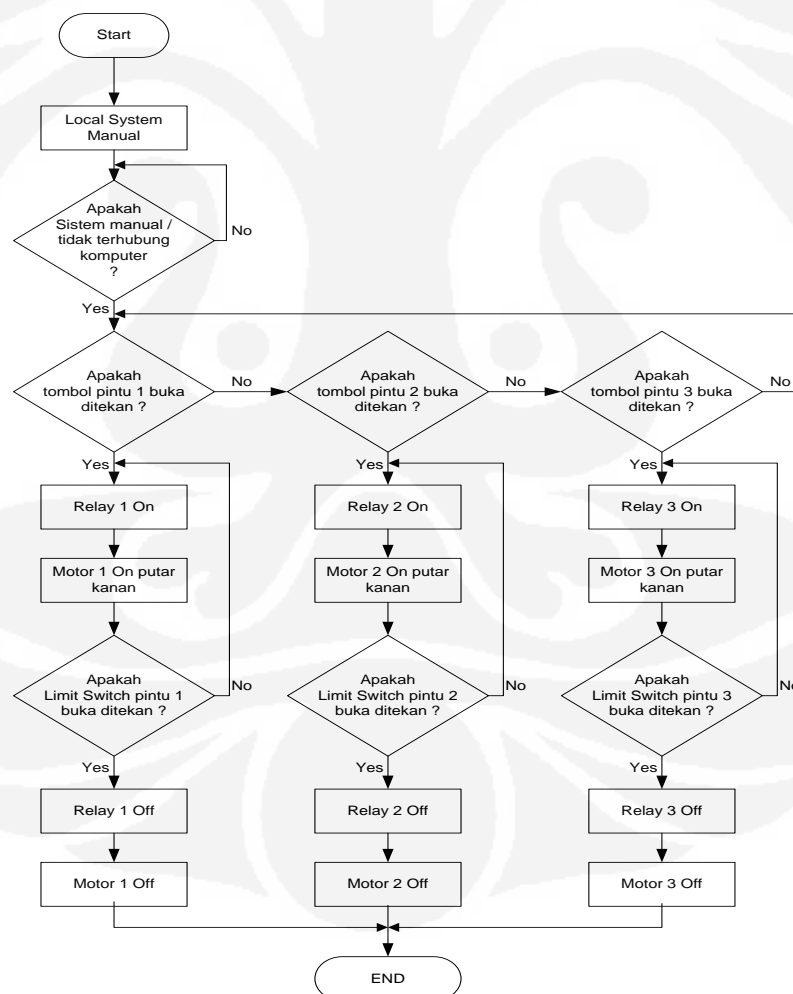
Dari gambar *ladder diagram* diatas terdapat dua *input* yaitu 0.00 dan 0.01 dimana *input* 0.00 adalah *input* untuk sensor naik dan 0.01 adalah *input* turun. Jika salah satu *input* tersebut aktif maka jumlah isi dari memori D0 akan berubah bertambah atau berkurang sesuai dengan *input* yang aktif. Setelah itu isi dari D0 akan di bandingkan dengan nilai-nilai tertentu yang



digunakan sebagai referensi untuk batasan buka dan tutup pintu. Untuk *output* yang di gunakan dapat dilihat pada Tabel 3.2

### ➤ *Mode Local dan Mode Remote*

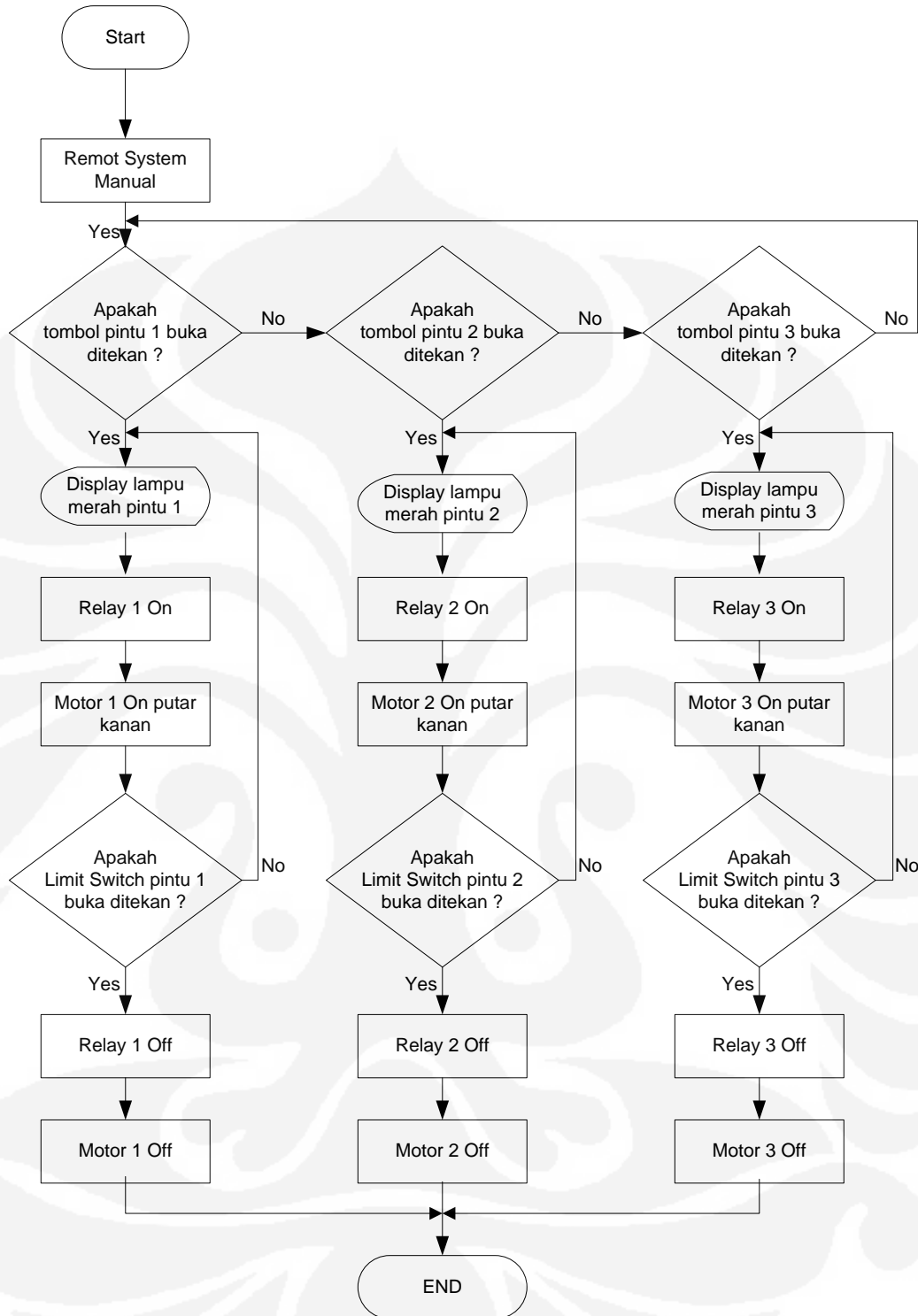
*Flowchart* untuk cara kerja mode *local* dan *remote* di bawah ini terdapat pada HMI, dalam pengoperasiannya kedua mode ini pada dasarnya sama yaitu pengontrolan pintu pada *plant* tanpa mengandalkan sensor untuk pergerakannya, jadi untuk mode *local* pengontrolan bebannya melalui *push button* yang ada di *plant*, sedangkan pada mode *remote* pengontrolan bebannya melalui HMI yang ada di komputer. Berikut adalah *flowchart* dari kedua mode tersebut:



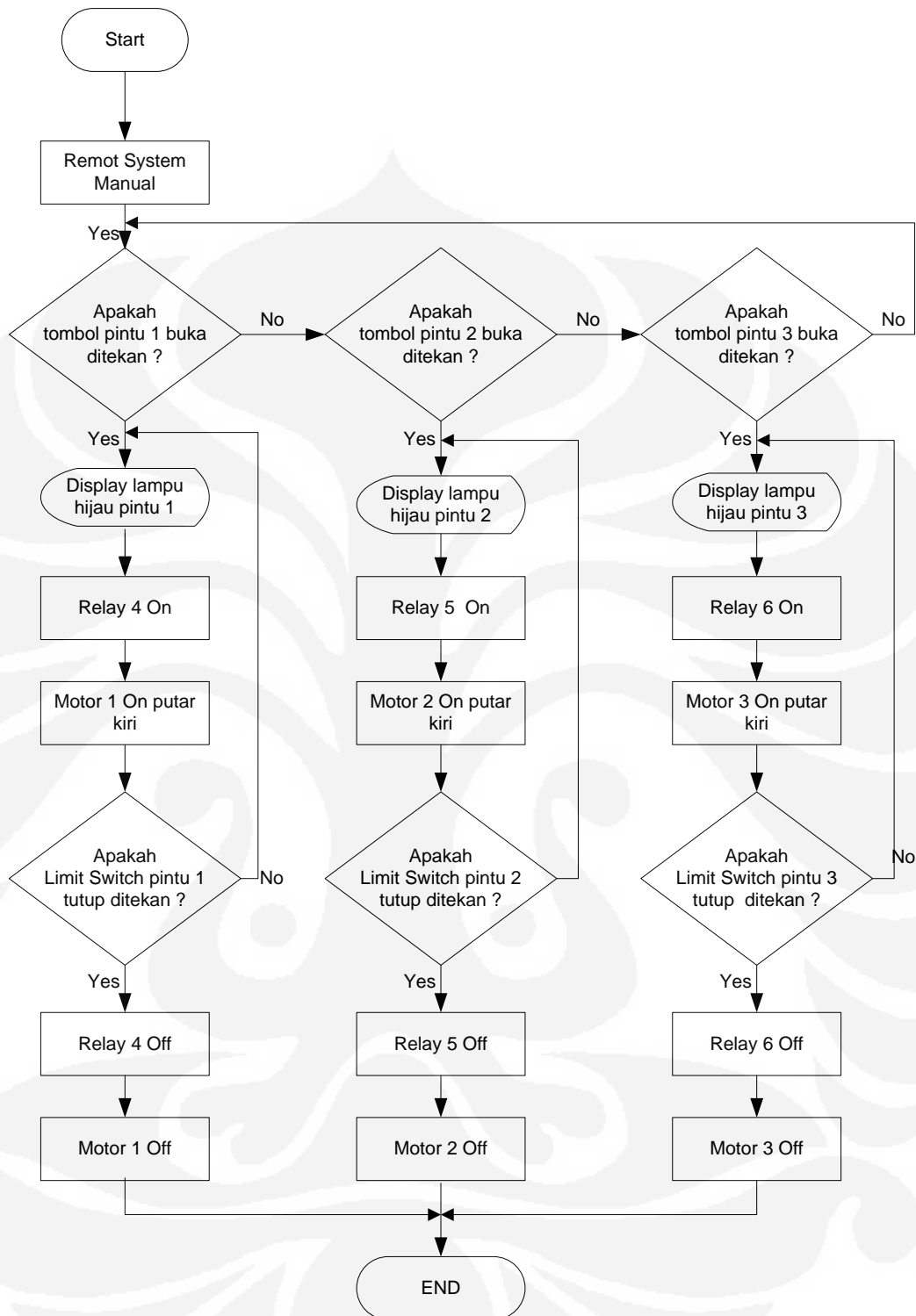
Gambar 3.18 *Flowchart Local System* Proses Buka



Gambar 3.19 Flowchart Local System Proses Tutup

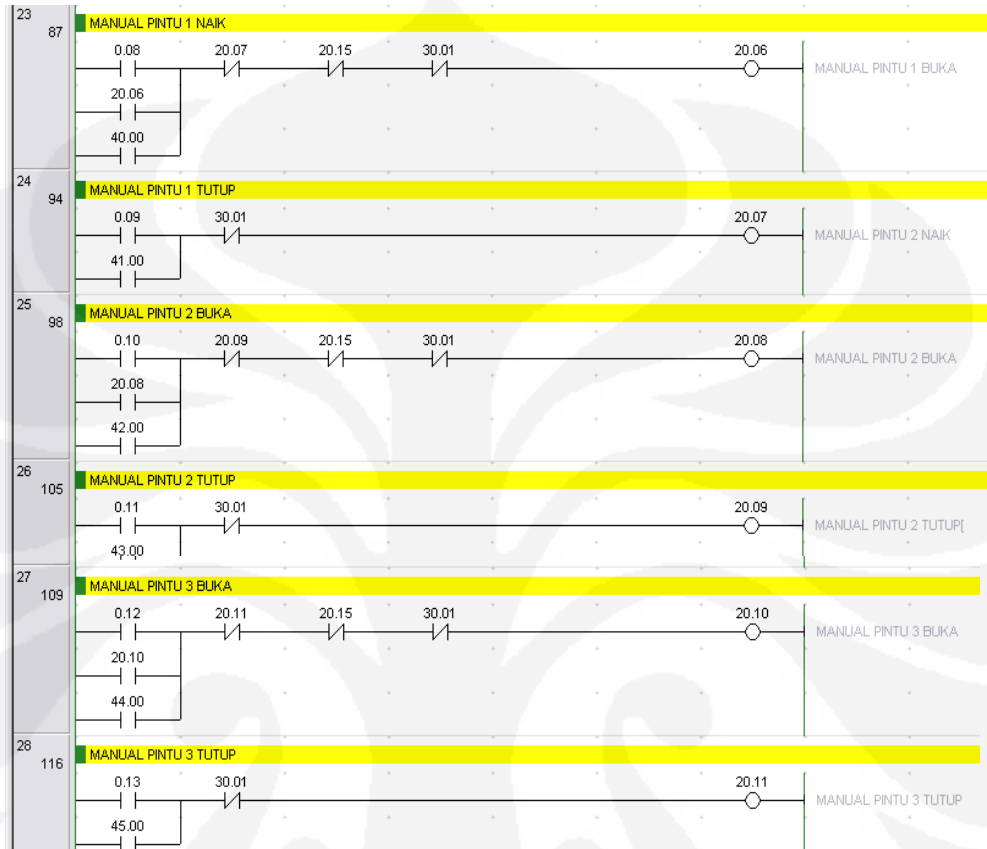


Gambar 3.20 *Flowchart Remote System Buka*



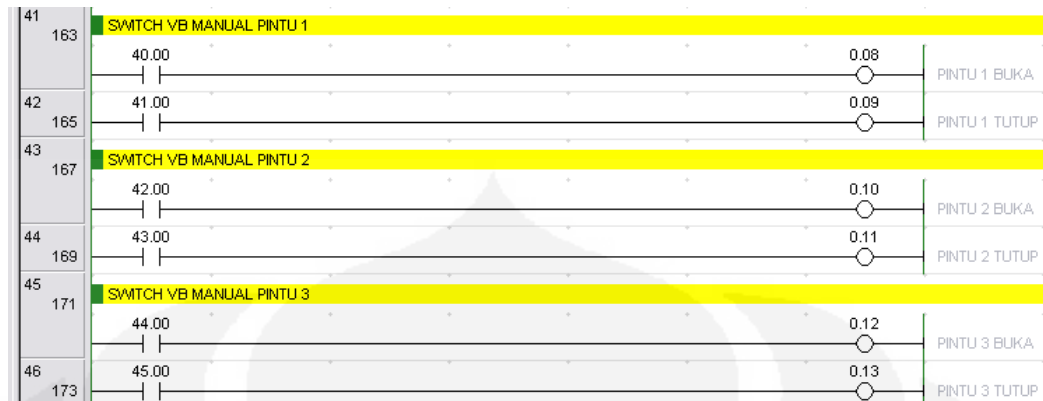
Gambar 3.21 *Flowchart Remote System Tutup*

Setelah mengetahui cara kerja dari diagram alir mode *local* dan *remote* maka langkah selanjutnya adalah membuat *ladder* yang akan di program ke PLC :



Gambar 3.22 *Ladder Mode Local*

Pada *ladder* di atas dapat dilihat bahwa *input* untuk membuka atau menutup pintu dilakukan dari tombol yang terdapat pada *panel*. *Input* yang di gunakan pada mode ini dapat di lihat pada tabel 3.2



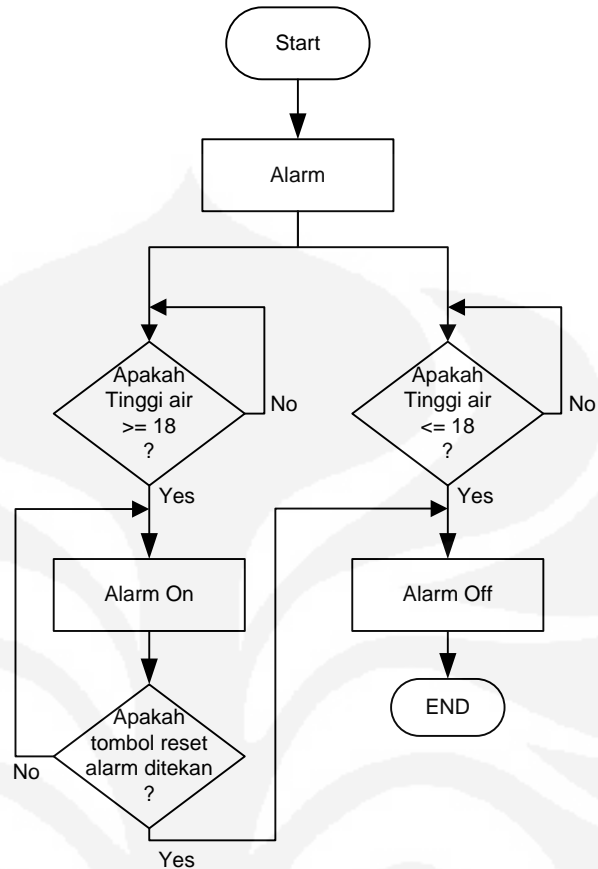
Gambar 3.23 *Ladder Mode remote*

Pada *ladder* di atas dapat dilihat bahwa *input* untuk membuka atau menutup pintu dilakukan dari tombol yang terdapat pada HMI komputer yang terhubung pada tombol *panel*. Tombol pada HMI komputer beralamat 40.00 sampai 45.00

#### ➤ *Alarm*

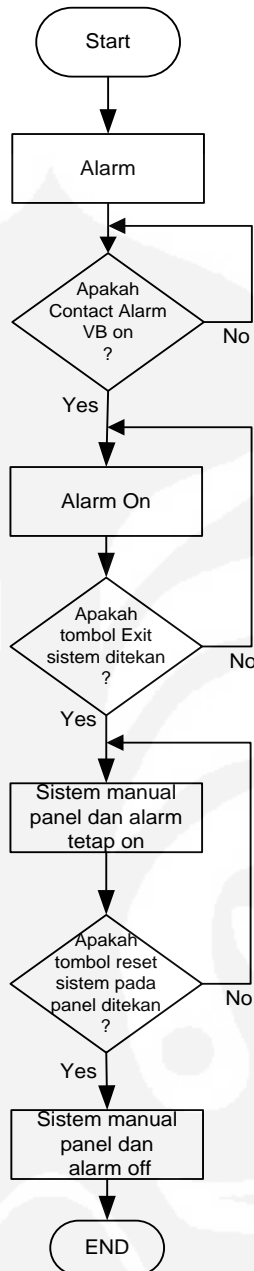
Keamanan yang dirancang pada sistem ini ada dua, yaitu system keamanan untuk antisipasi banjir dan yang ke-dua yaitu system pemberitahuan apabila terjadi masalah pada pintu banjir kanal, seperti pintu tidak bias buka atau tutup maksimal atau tidak lancar karena terhambat kotoran. Jika salahsatu system keamanan ini aktif maka LED alarm dan *Buzzer* akan *on* dan *off* secara begantian.

Untuk mempermudah dalam merancang system alarm maka terlebih dahulu dibuat *flow chart* untuk system yang akan di buat. Berikut adalah *flow chart* untuk system alarm antisipasi banjir:



Gambar 3.24 *Flowchart* Alarm Antisipasi Banjir

Begitu juga pada alarm gangguan pintu agar mudah dalam merancang system alarm jika ada gangguan pada pintu banjir kanal, dimana system ini akan dikendalikan oleh program HMI komputer maka terlebih dahulu dibuat *flow chart* untuk system yang akan di buat. Berikut adalah *flow chart* untuk system alarm jika terjadi gangguan pada pintu banjir kanal:

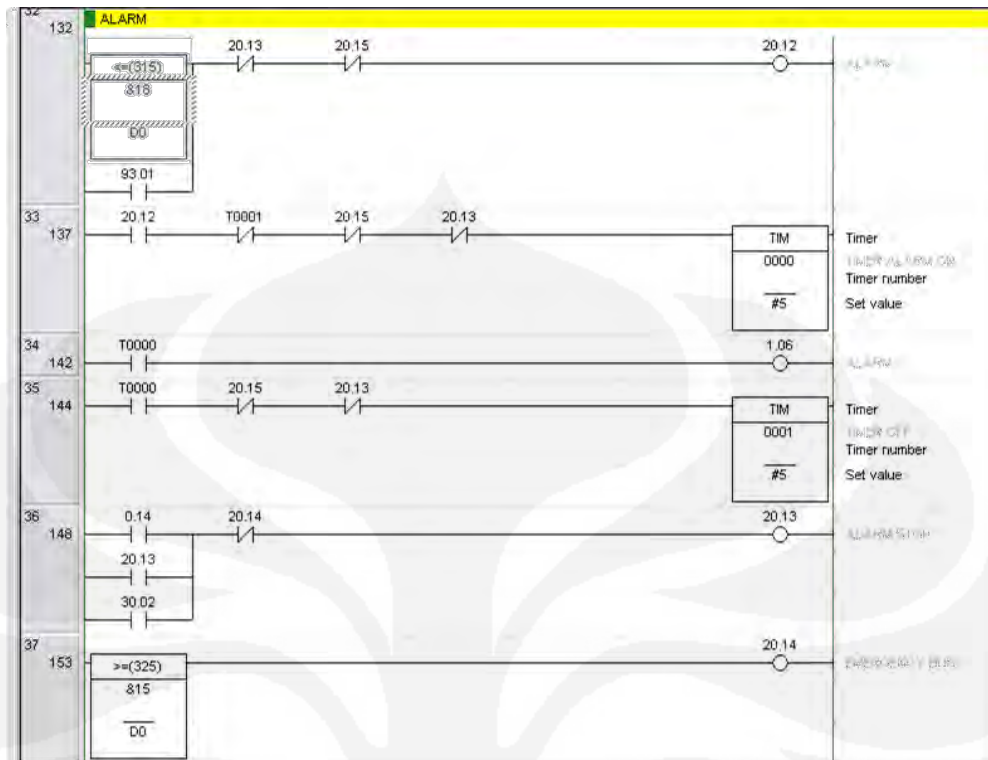


Gambar 3.25 *Flowchart* Alarm Gangguan Pintu Air

Setelah membuat *flowchart* dari kedua jenis alarm *system* maka dapat diaplikasikan kedalam bentuk *ladder* pada pemrograman PLC.

Berikut adalah *ladder* untuk *Alarm* antisipasi banjir, yang akan berfungsi jika air telah melewati batas ketinggian yang telah di tentukan.





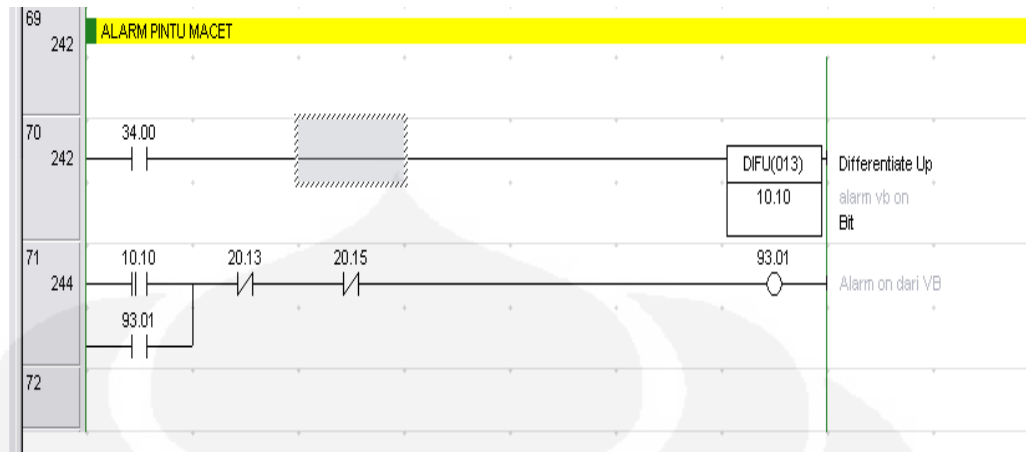
Gambar 3.26 *Ladder Alarm*

Dari gambar *ladder diagram Alarm* diatas dapat dilihat bahwa alarm akan aktif jika isi dari D0 adalah lebih besar sama dengan 18, jika hal ini terpenuhi maka TIM 0000 untuk timer 0 dan TIM 0001 akan nyala bergantian dengan lama nyala sekitar setengah detik.

*Output* untuk alarm ini yaitu 1.06 yang di hubungkan ke LED dan *Buzzer*.

Selain itu untuk menyalakan alarm dapat juga di aktifkan dari program *Visual Basic* yang akan mengaktifkan *alarm* jika terjadi gangguan pada pintu seperti naik atau turun pintu tidak lancar atau macet.

Pada gambar di bawah ini dapat di lihat bahwa contact 34.00 akan mengaktifkan *output* 93.01 yang merupakan kontak yang di pakai untuk mengaktifkan alarm jika terjadi gangguan pada pintu.



Gambar 3.27 Contact Alarm dari Visual Basic

### 3.3.2 Perancangan Software HMI

*Human Machine Interface* (HMI) merupakan bagian yang ada dalam sistem SCADA, HMI menampilkan gambar animasi *plant* yaitu sebuah Banjir kanal yang terdiri dari unit-unit beban yang akan di kontrol. HMI ini digunakan untuk memonitoring dan mengontrol *plant* dari jarak jauh. Untuk kode program pada masing-masing *form* bisa dilihat di lampiran.

#### 3.3.2.1 Form Main Program

Pada *form main program* ini akan ditampilkan simulasi dari *plant*. Pada *form* ini terdapat bagian untuk *monitoring* dan *controlling* unit-unit beban yang ada pada *plant*.



Gambar 3.28 Form Main Program

### 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 tugas akhir 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” dan “Adodc2”.



Gambar 3.29 ADODC

## BAB 4 PENGUJIAN SISTEM

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Pengujian ini meliputi pengujian software HMI SCADA yaitu untuk sistem *auto* atau *manual* dengan mode *local* dan *remote*. Serta pengujian hardware seperti sensor ketinggian air dan pengujian motor penggerak pintu,

### 4.1 Pengujian Mode *Auto*

Pada mode ini dilakukan pengujian berapa lama waktu yang dibutuhkan HMI SCADA dalam memberikan informasi ketinggian air pada tabung pengukur. Cara kerja pada sistem *auto* ini adalah mengisi tabung pengukur dengan air sehingga pelampung naik dan memutar piringan pada sensor ketinggian air. Ketika air mulai naik maka keterangan tinggi air HMI SCADA pada computer akan berubah sesuai dengan ketinggian air pada tabung pengukur. pada pemrograman *Visual Basic* digunakan *timer 2* dengan *set interval time* adalah 100 ms, sebagai fungsi *scanning* untuk ketinggian air sedangkan data *memory* (DM) yang ada pada PLC Omron CJ1M di gunakan sebagai input *scanning* pada *Program*

1. Berikut ini data pengujian 1 yang di dapat dengan pengaturan :

➤ *Timer 2 ; Interval Time : 100 ms*

Tabel 4.1 Hasil Pengujian 1 Program Deteksi Ketinggian Air

Unit Beben	Waktu (s) (Ave)	
	HMI	Miniatur
Pintu 1 Naik	4.9	0.2
Pintu 1 Turun	5.6	0.2

Kemudian dari pengujian program *system auto* dinyatakan berhasil 100% sesuai deskripsi.

## 4.2 Pengujian mode *Local*

Pada mode ini dilakukan pengujian dengan menekan tombol push button yang ada pada miniatur banjir kanal, lalu diuji berapa lama waktu yang dibutuhkan hingga pintu-pintu pada HMI SCADA pada komputer bergerak naik atau turun, dan diuji juga berapa waktu yang diperlukan pintu untuk naik maksimum dan turun maksimum pada miniatur banjir kanal maupun di HMI SCADA pada computer. Deskripsi kerja pada mode *local* ini yaitu ketika tombol push button pada miniatur banjir kanal ditekan maka pintu-pintu pada miniature banjir kanal akan naik atau turun serta lampu-lampu indicator di HMI SCADA pada komputer akan berubah seperti pada saat pintu naik maka lampu hijau dan jika pintu turun maka lampu indicator akan menyala merah, pada pemrograman Visual Basic digunakan *timer* 4,6,8 sebagai fungsi scanning untuk pintu naik dan *timer* 5,7,9 sebagai fungsi scanning untuk pintu turun sedangkan data memori (DM) yang ada pada PLC Omron CJ1M di gunakan sebagai input scanning pada Visual Basic.

Pengujian waktu untuk pintu pada program HMI SCADA yaitu dengan *Set Interval Time Timer* 4 sampai *timer* 9 adalah 10 ms hal ini dikarenakan program HMI SCADA akan berat dan sering hang jika menggunakan waktu scanning yang terlalu cepat. Berikut adalah pengujian yang dilakukan:

1. Pengujian 1 yaitu pengujian respon pada *plant* maupun pada HMI SCADA komputer ketika tombol *open* atau *close* ditekan. Tabel di bawah adalah data pengujian yang di dapat dengan pengaturan :

➤ *Timer 4,5,6,7,8,9 ; Interval Time : 20 ms*

Tabel 4.2 Hasil Pengujian 1 Respon Dari Tombol Panel

Unit Beben	Waktu (s) (Ave)	
	HMI	Miniatur
Pintu 1 Naik	3.1	0.3
Pintu 1 Turun	3.3	0.5
Pintu 2 Naik	2.7	0.3
Pintu 2 Turun	2.3	0.5
Pintu 3 Naik	2.3	0.4
Pintu 3 Turun	2.5	0.5

Pengujian 2 yaitu pengujian lama pergerakan pintu dari keadaan tutup sampai naik maksimum dan dari buka maksimum sampai tutup maksimum pada *plant* maupun pada HMI SCADA dari tutup hingga buka maksimum dan dari buka hingga tutup maksimum. Tabel di bawah adalah data pengujian yang didapat dengan pengaturan :

➤ *Timer 4,5,6,7,8,9 ; Interval Time : 20 ms*

Tabel 4.3 Hasil Pengujian 2 Kecepatan Buka Tutup Pintu pada mode *local*

Unit Beben	Waktu (s) (Ave)	
	HMI	Miniatur
Pintu 1 Naik	25.2	14.2
Pintu 1 Turun	25.7	23.0
Pintu 2 Naik	25	11.8
Pintu 2 Turun	24.7	23.1
Pintu 3 Naik	24.3	13.0
Pintu 3 Turun	24.3	22.6

Dari kedua tabel di atas dapat dilihat perbedaan waktu respon pada HMI SCADA komputer ketika tombol *open* atau *close* ditekan dari panel, hal ini dikarenakan pada program HMI SCADA komputer melakukan *scanning* pintu dari yang membutuhkan waktu lebih lambat dibandingkan dengan waktu *scanning* pada PLC. Kemudian dari hasil pengujian tersebut program mode *local* berhasil 100% sesuai deskripsi

### 4.3 Pengujian mode *Remote*

Pada mode ini dilakukan pengujian dengan menekan tombol *open* atau *close* yang ada di HMI SCADA, lalu dilakukan pengujian yang sama dengan mode *local*. Deskripsi kerja pada mode *local* ini yaitu ketika tombol *push button* di HMI SCADA pada komputer ditekan maka pintu-pintu pada miniature dan di HMI SCADA pada komputer akan bergerak naik atau turun serta lampu-lampu indicator di HMI SCADA pada komputer akan berubah seperti pada saat pintu naik maka lampu hijau dan jika pintu turun maka lampu indicator akan menyala merah. Pada mode *remote* ini juga dilakukan pengujian terhadap tombol naik turun pada saat ingin men-*setting* ketinggian air dan juga pada mode ini akan dilakukan terhadap respon *CCTV* ketika tombol *CCTV* pada HMI SCADA di tekan, Berikut ini adalah hasil pengujian yang dilakukan pada mode *remote* :

1. Pengujian 1 yaitu pengujian respon pada *plant* maupun pada HMI SCADA ketika tombol pada HMI SCADA ditekan. Tabel di bawah adalah data pengujian yang di dapat dengan pengaturan :

➤ *Timer* 4,5,6,7,8,9 ; *Interval Time* : 20 ms

Tabel 4.4 Hasil Pengujian 1 Respon Dari Tombol HMI SCADA

Unit Beben	Waktu (s) (Ave)	
	HMI	Miniatur
Pintu 1 Naik	3.4	0.5
Pintu 1 Turun	3.1	0.6
Pintu 2 Naik	3.0	0.6
Pintu 2 Turun	2.0	0.9
Pintu 3 Naik	2.8	0.5
Pintu 3 Turun	3.0	0.9

2. Pengujian 2 yaitu pengujian lama pergerakan pintu pada *plant* maupun pada HMI SCADA dari tutup hingga buka maksimum dan dari buka hingga tutup maksimum. Tabel di bawah adalah data pengujian yang di dapat dengan pengaturan :

➤ *Timer* 4,5,6,7,8,9 ; *Interval Time* : 20 ms

Tabel 4.5 Hasil Pengujian 2 Kecepatan Buka Tutup Pintu pada mode *remote*

Unit Beben	Waktu (s) (Ave)	
	HMI	Miniatur
Pintu 1 Naik	26.6	14.0
Pintu 1 Turun	35.3	22.7
Pintu 2 Naik	24.7	11.4
Pintu 2 Turun	26.0	23.2
Pintu 3 Naik	24.9	13.6
Pintu 3 Turun	24.9	23.5

3. Pengujian 3 yaitu pengujian respon pada HMI SCADA ketika tombol *set water* untuk mengatur ketinggian air pada HMI SCADA sehingga pembacaan pada HMI SCADA dapat dilakukan dengan benar sesuai dengan ketinggian pada *plant*. Tabel di bawah adalah data pengujian yang di dapat :

Tabel 4.6 Hasil Pengujian Respon Tombol Setting Water HMI SCADA

Unit Beben	Waktu (s) (Ave)
	HMI
Tombol Naik	0.1
Tombol Turun	0.1

4. Pengujian 4 yaitu pengujian respon pada *plant* maupun pada HMI SCADA ketika tombol *CCTV* pada HMI SCADA ditekan.

Tabel di bawah adalah data pengujian yang di dapat :



Tabel 4.7 Hasil Pengujian Respon Program *CCTV*

Beban	Waktu (s) (Ave)
	HMI
<i>CCTV</i>	1

Dari tabel 1 dan 2 dapat dilihat perbedaan waktu pada HMI SCADA komputer ketika tombol *open* atau *close* ditekan hal ini dikarenakan pada program SCADA komputer melakukan *scanning* pintu dari PLC yang membutuhkan waktu lebih lambat dibandingkan dengan waktu pada PLC.

Sedangkan dari tabel 3 dan 4 dapat dilihat bahwa waktu yang di butuhkan cukup cepat hal ini dikarenakan program pada HMI SCADA komputer tidak harus melakukan proses *scanning* dari PLC.

Hasil pengujian tersebut program mode *remote* berhasil 100% sesuai deskripsi.

#### 4.4. Pengujian Alarm

Pada pengujian alarm ini ada dua macam pengujian, yang pertama adalah pengujian alarm untuk antisipas banjir dan yang ke-dua yaitu pngujian untuk gangguan pada pintu air.

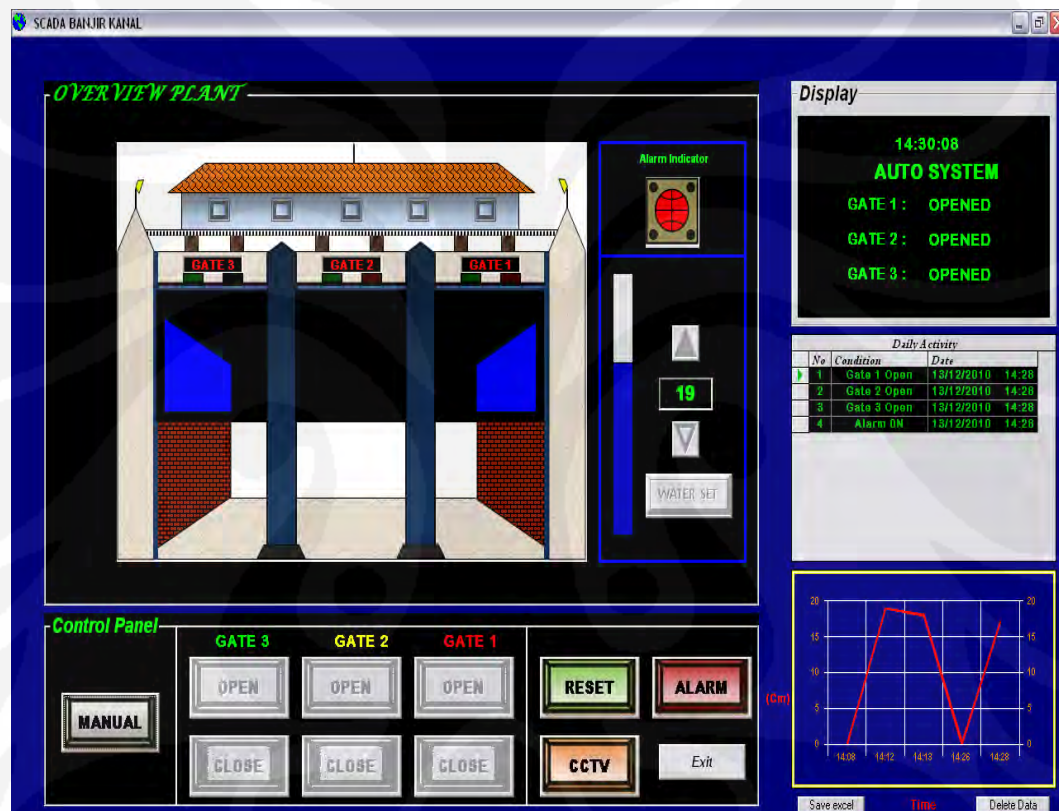
##### ➤ Pengujian Alarm Antisipasi Bajir

Pengujian dilakukan dengan dua cara, cara yang pertama yaitu dengan menaikan pelmpung sehingga piringan sensor ketinggian air akan berputar dan sensor mendeteksi ketinggian air atau dengan menggunakan mode *manual* pada HMI SCADA komputer dan selanjutnya mengatur ketinggian air dengan menekan tombol *up* sampai batas yang akan membuat alarm menyala dan setelah itu memindahkan mode ke mode *auto*.

Batas ketinggian air untuk mengaktifkan alarm antisipasi banjir adalah lebih besar sama dengan 18. Jadi apabila ketinggian air lebih besar sama dengan 18 maka alarm antisipasi banjir akan aktif.

Berikut adalah hasil pengujian yang dilakukan pada alarm antisipasi banjir:

1. Pengujian 1 yaitu dengan memutar piringan sehingga terbaca ketinggian air naik hingga lebih besar sama dengan 18 :



Gambar 4.1 Tampilan Alarm Hidup Dengan Pembacaan Sensor Ketinggian Air

2. Pengujian 2 yaitu dengan menggunakan menekan tombol *up* pada HMI SCADA komputer dari mode *manual* :



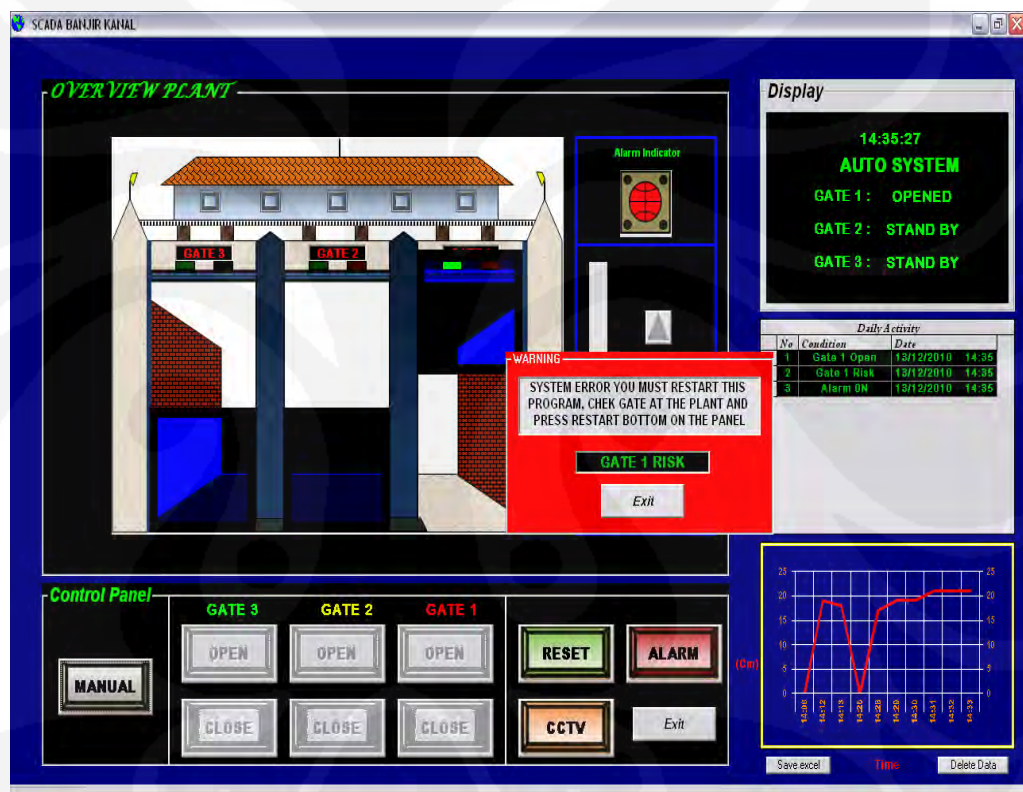
Gambar 4.2 Alarm Hidup Dengan Menaikan Ketinggian Air Secara *Manual*

Waktu yang di butuhkan hingga alarm *on* pada saat air mencapai ketinggian 18 adalah rata-rata 0.2 detik baik pada mode *Auto* maupun *manual*. Pada *plant buzzer* dan LED akan *on* dan *off* bergantian sedangkan pada HIM SCADA komputer waktu yang di butuhkan hingga tanda alarm hidup adalah rata-rata 0,4 detik pada hal ini hanya indikator alarm yang nyala dan mati bergantian, dan pembacaan pada tabel condition akan tercatat *Alarm On*.

Dari hasil pengujian tersebut program untuk alarm antisipasi banjir berhasil 100% sesuai deskripsi.

### ➤ Pengujian Alarm Gangguan Pintu

Pengujian dilakukan dengan men-setting timer pada *interval time* yang digunakan pada pintu yang akan di uji, dalam pengujian ini yang di-set adalah timer 4 untuk pengujian pintu 1. *Interval Timer* pada pintu 1 dirubah agar pergerakan naik di HMI SCADA komputer lebih cepat di bandingkan pintu sebenarnya yang ada di *plant*, sehingga pintu pada HMI SCADA komputer akan bergerak melebihi batas maksimum pintu naik. Berikut adalah hasil pengujian yang dilakukan pada alarm antisipasi banjir:



Gambar 4.3 Tampilan Alarm Hidup Karena Ada Kerusakan Pada Pintu 1



Gambar 4.4 Tampilan Tanda Peringatan Jika Kerusakan Pada Salah Satu Pintu

Dari tampilan dapat dilihat jika terjadi gangguan pada pintu maka pintu yang yang terjadi gangguan akan melewati batas atas atau bawah pergerakan pintu. Waktu yang di butuhkan hingga alarm *on* pada terjadi gangguan pada salah satu pintu adalah rata-rata 0.1 detik baik pada mode *Auto* maupun *Manual*. Pada *plant buzzer* dan LED akan *on* dan *off* bergantian sedangkan pada HIM SCADA komputer waktu yang di butuhkan hingga tanda alarm hidup adalah rata-rata 0,4 detik dalam hal ini indicator alarm yang nyala dan mati bergantian dan aada tanda peringatan untuk keluar dari system, serta pembacaan pada tabel condition akan tercatat pintu mana yang mengalami gangguan setelah itu dibawahnya tercatat Alarm *On*.

Dari hasil pengujian tersebut program untuk alarm gangguan pintu berhasil 100% sesuai deskripsi.

#### 4.5 Pengujian Database

Pengujian yang dilakukan pada database adalah melihat apakah setelah program SCADA dijalankan maka program database pada *Microsoft Access* dapat berjalan dengan baik.

1. Pengujian 1 yaitu pengujian untuk program penyimpanan database kondisi plant, dimana pengujian ini dengan menggunakan pengaturan :



- *Set timer* : 11,12,13,14,15,16 dengan *interval time* 20000

Hasil dari pengujian tersebut dapat di lihat pada gambar di bawah ini:

No	Condition	Date
1	Gate 1 Open	17/12/2010 03:06
2	Gate 2 Open	17/12/2010 03:15
3	Gate 2 Close	17/12/2010 03:18
4	Gate 2 Open	17/12/2010 03:20
5	Gate 3 Open	17/12/2010 03:32
6	Gate 3 Close	17/12/2010 03:34
7	Gate 2 Close	17/12/2010 03:35
8	Gate 1 Close	17/12/2010 03:36

Gambar 4.5 Database Untuk Kondisi Plant

Dari hasil pengujian tersebut dapat dilihat bahwa data akan bertambah jika ada perubahan kondisi saja, dengan berjalannya program penyimpanan data kondisi plant maka program ini dinyatakan berhasil 100% sesuai deskripsi

2. Pengujian 2 yaitu pengujian untuk program penyimpanan database kondisi Air pada plant, dimana pengujian ini dengan menggunakan pengaturan :

- *Set timer* : 18 dengan *interval time* 60000

Hasil dari pengujian tersebut dapat di lihat pada gambar di bawah ini:

No	Height	Time
1	0	03:43
2	1	03:44
3	1	03:45
4	1	03:46
5	1	03:47
6	1	03:48
7	3	03:49
8	6	03:50
9	4	03:51

Gambar 4.6 Database Ketinggian Air Pada Plant

Dari hasil pengujian tersebut dapat dilihat bahwa data akan bertambah dengan perubahan waktu 1 menit. Maka dengan berjalannya program penyimpanan data kondisi air pada plant maka program ini dinyatakan berhasil 100% sesuai deskripsi

- Pengujian 3 yaitu pengujian untuk program penyimpanan database kondisi Air pada plant, dimana pengujian ini dengan menggunakan database pada *Microsoft Excel* :

	A	B	C
1		<b>DATA WATER LEVEL</b>	
2	<b>No</b>	<b>Water Lev</b>	<b>Time</b>
3	1	0	03:43
4	2	1	03:44
5	3	1	03:45
6	4	1	03:46
7	5	1	03:47
8	6	1	03:48
9	7	3	03:49
10	8	6	03:50
11	9	4	03:51

Gambar 4.7 Database Ketinggian Air Pada Microsoft Excel

Dengan berjalannya program penyimpanan data kondisi ketinggian air pada *Microsoft Excel* maka program ini dinyatakan berhasil 100% sesuai deskripsi

#### 4.6 Pengujian Sensor Ketinggian Air

Pengujian yang dilakukan terhadap sensor ketinggian air adalah untuk mengetahui seberapa cepat respon yang dapat diterima oleh sensor terhadap kenaikan air. Untuk mengetahui respon sensor tersebut dapat dilakukan pengujian dengan dua cara. Cara yang pertama dengan mengisi tabung pengukur dengan air melalui tabung pengisi dan amati berapa waktu yang dibutuhkan untuk menaikkan atau menurunkan pelampung dengan satuan Cm. cara yang kedua yaitu dengan menarik bandul pemberat dengan kecepatan yang telah di tentukan. Setelah cara kedua tersebut dilakukan maka akan didapat waktu tercepat yang masih bias di respon oleh sensor ketinggian air.

- Berikut adalah tabel data hasil pengujian untuk sensor ketinggian air dengan cara 1:

Tabel 4.8 Hasil Pengujian Sensor Ketinggian Air Cara 1

Arah Air	Kecepatan Air (s / Cm) (Ave)	Respon Sensor
Naik	5.5	Baik
Turun	5.0	Baik



- Berikut adalah tabel data hasil pengujian untuk sensor ketinggian air dengan cara 2:

Tabel 4.9 Hasil Pengujian Sensor Ketinggian Air Cara 2

Kecepatan Air (s / Cm) (Ave)	Respon Sensor
5.0	Baik
4.5	Baik
4.0	Baik
3.5	Baik
3.0	Baik
2.5	Baik
2.0	Baik
1.5	Baik
1.0	Kurang Baik
<1.0	Tidak Merspon

Dari hasil pengujian tersebut maka sensor ketinggian air berhasil 100% sesuai deskripsi karena sensor mampu merspon lebih cepat dari kecepatan normal masukan air kedalam tabung pengukur.

## BAB 5

### KESIMPULAN

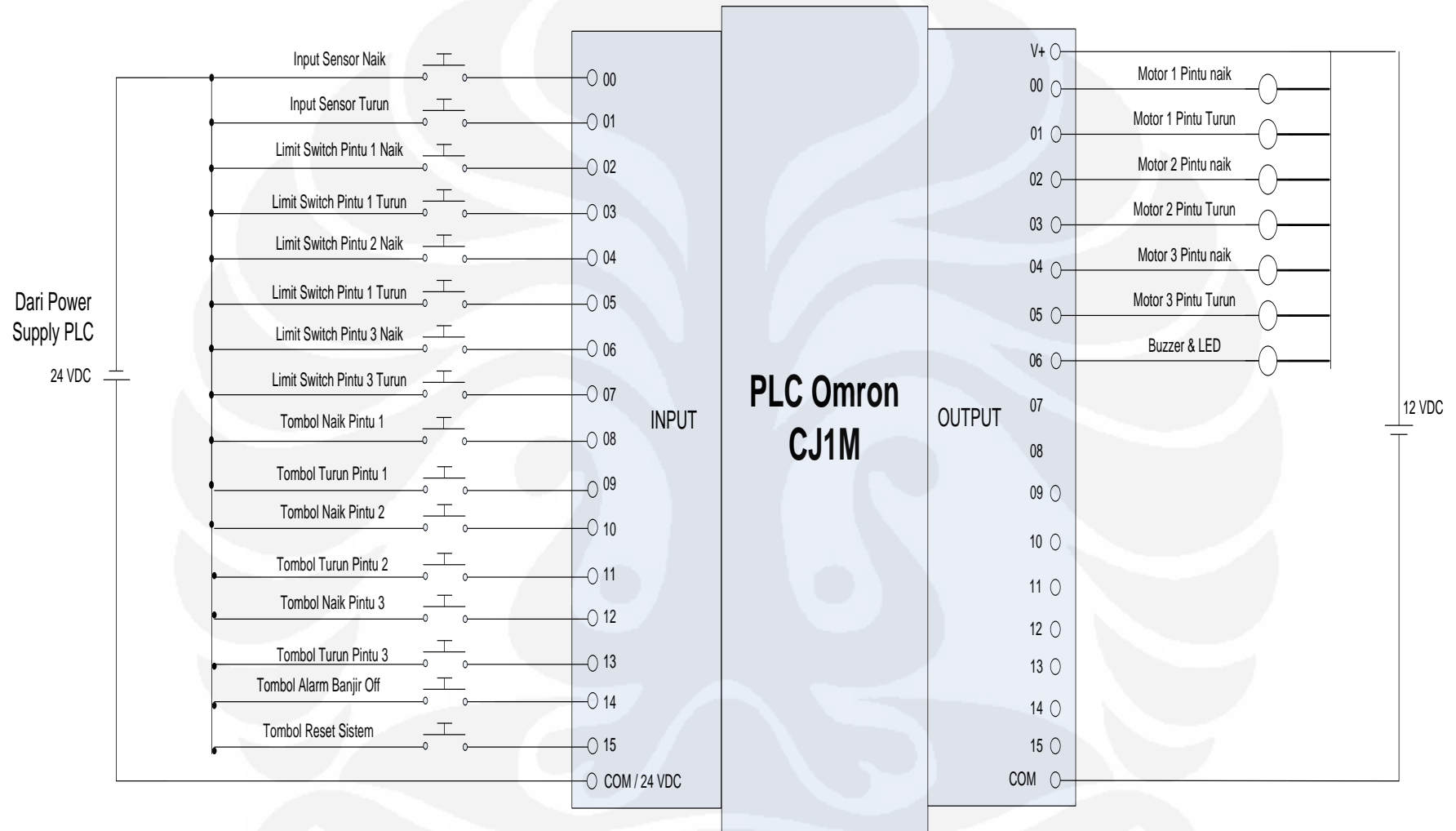
1. Pada pengujian dengan system otomatis waktu respon yang diperlukan HMI SCADA untuk dapat menampilkan tinggi air yaitu rata-rata 4,9 detik untuk naik dan rata-rata 5,6 detik untuk turun sedangkan waktu respon pada Miniatur adalah 2,0 detik untuk naik dan turun.
2. Pengujian system manual respon waktu yang diperlukan mode *local* untuk menyalakan unit-unit beban pada miniatur pintu air adalah 0,3 sampai dengan 0,5 detik sedangkan respon waktu yang diperlukan mode *remote* untuk menyalakan unit-unit beban pada miniatur pintu air sedikit lebih lambat yaitu 0,5 sampai dengan 0.9 detik,
3. Program HMI SCADA yang dirancang merupakan program tetap terhadap miniatur plant pintu air yang dibuat, apabila ada penambahan device atau unit beban maka harus menambah *source code* pada program HMI SCADA-nya dan mendesain ulang tampilan HMI SCADA-nya.
4. Sensor memerlukan waktu *delay* sebesar  $\pm 0,7$  detik untuk mengaktifkan relay saat terjadi perubahan arah pada piringan piringan sensor, bila waktu perubahan arah pada piringan sensor ketinggian air dibawah 1 detik per cm maka dapat menjadikan sensor ketinggian air tidak merespon

## 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 *microcontroller*: Setiawan, rahmat. MIKROKONTROLER MCS-51. Yogyakarta: penerbit Graha Ilmu, 2006.
- [6] Dasar-dasar *Microsoft Access*,  
[http://id.wikipedia.org/wiki/Microsoft\\_Access](http://id.wikipedia.org/wiki/Microsoft_Access)
- [7] Dasar-dasar relay, <http://depokinstruments.com/2010/02/20/seulas-teori-relay/>
- [8] Dasar-dasar Motor DC,  
<http://staff.ui.ac.id/internal/040603019/material/DCMotorPaperandQA.pdf>
- [9] Dasar-dasar Limit switch,  
<http://repository.usu.ac.id/bitstream/123456789/18772/3/Chapter%20II.pdf>
- [10] Datasheet PLC Omron CJ1M CPU 11, CJ1W ID211, CJ1W OD211 dan Power Supply PA 202.



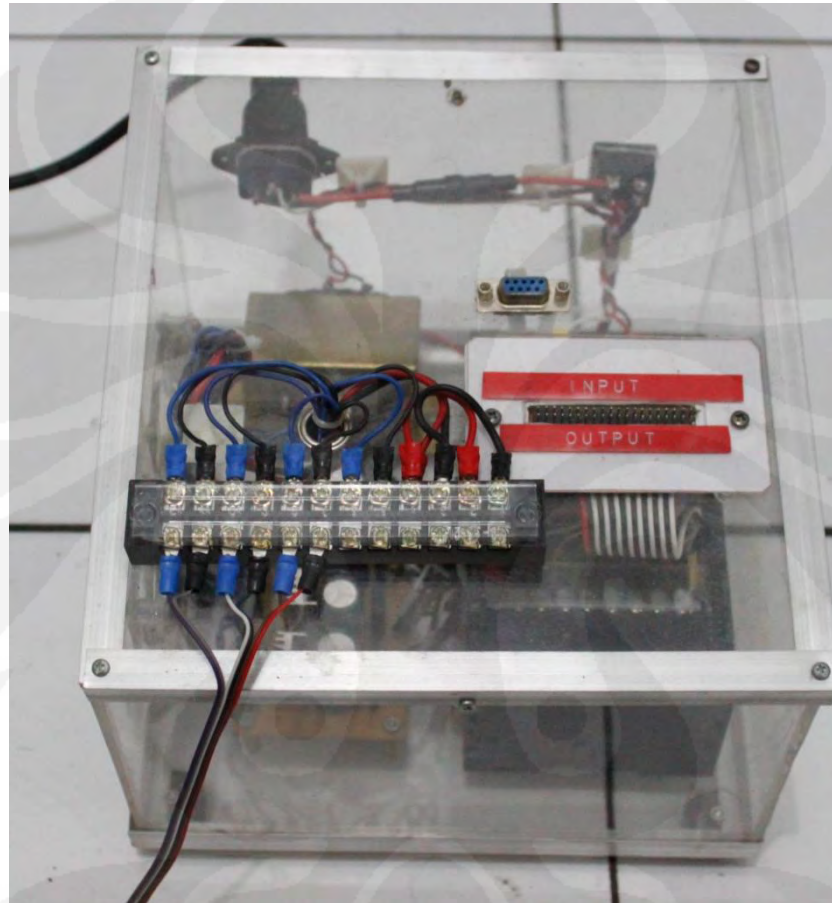
# LAMPIRAN 1



## Wiring Diagram PLC

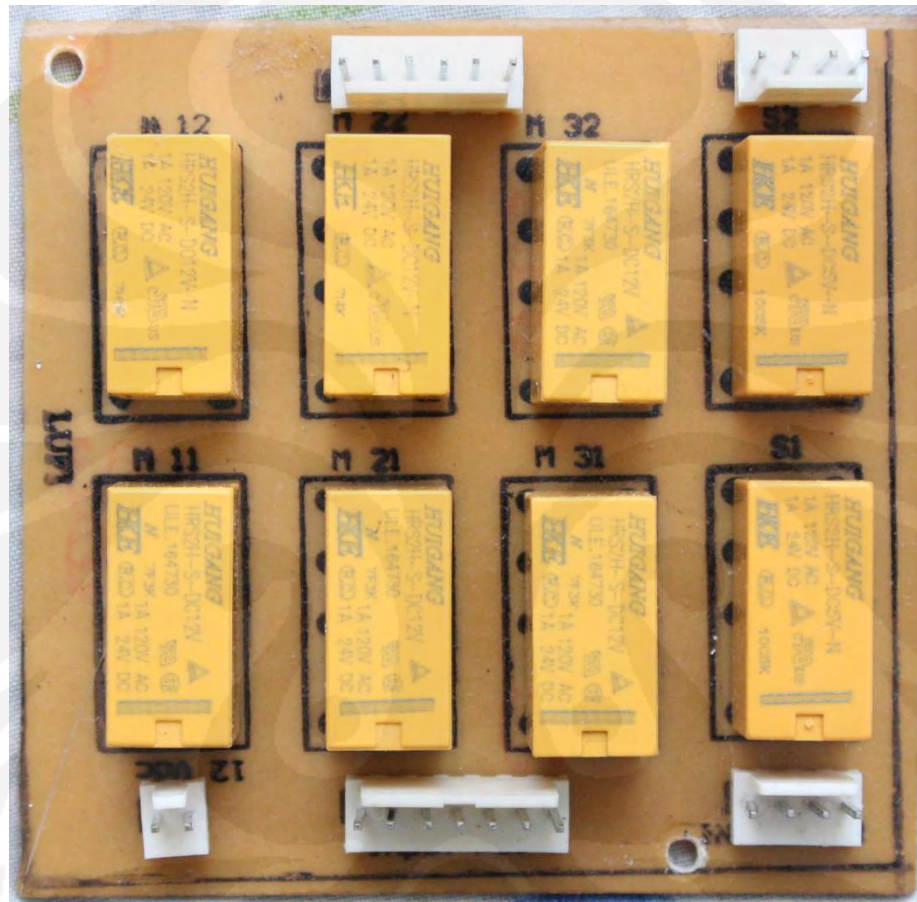


**Gambar Minatur Plant Pintu Air Banjir Kanal**



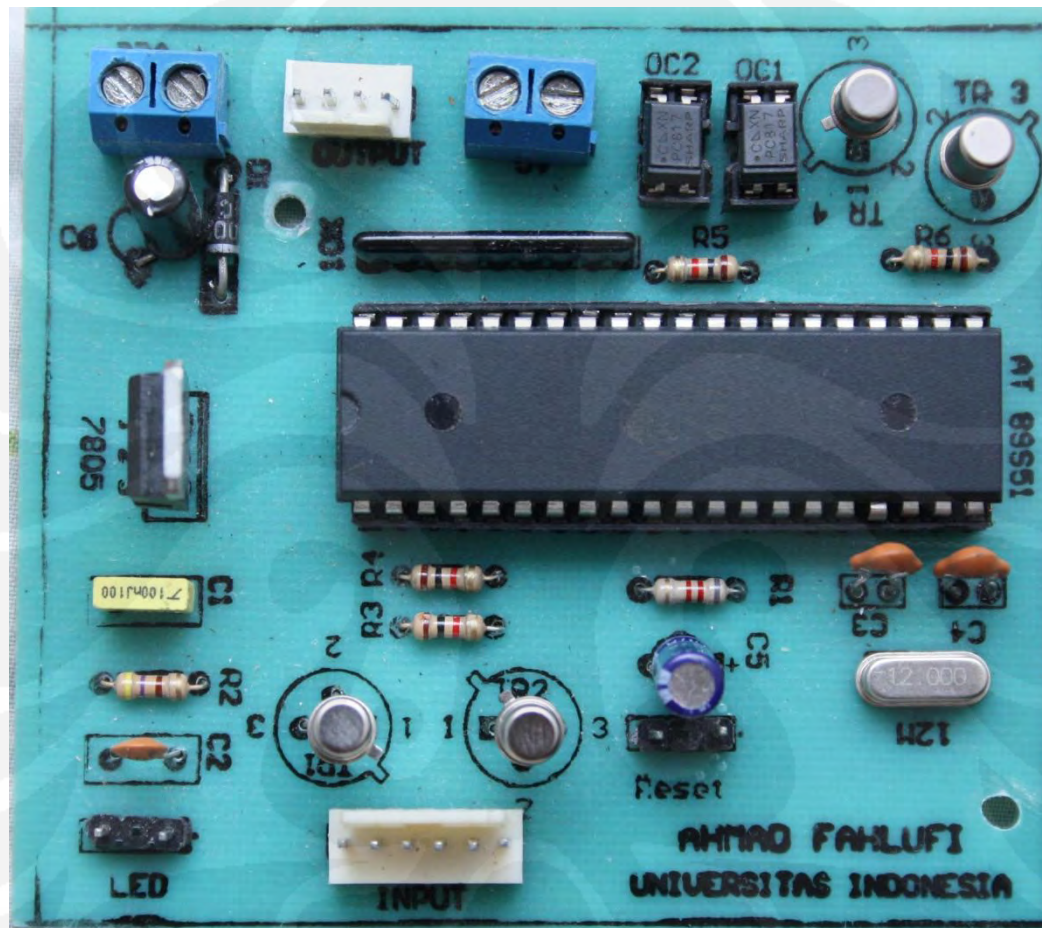
**Gambar Modul PLC**



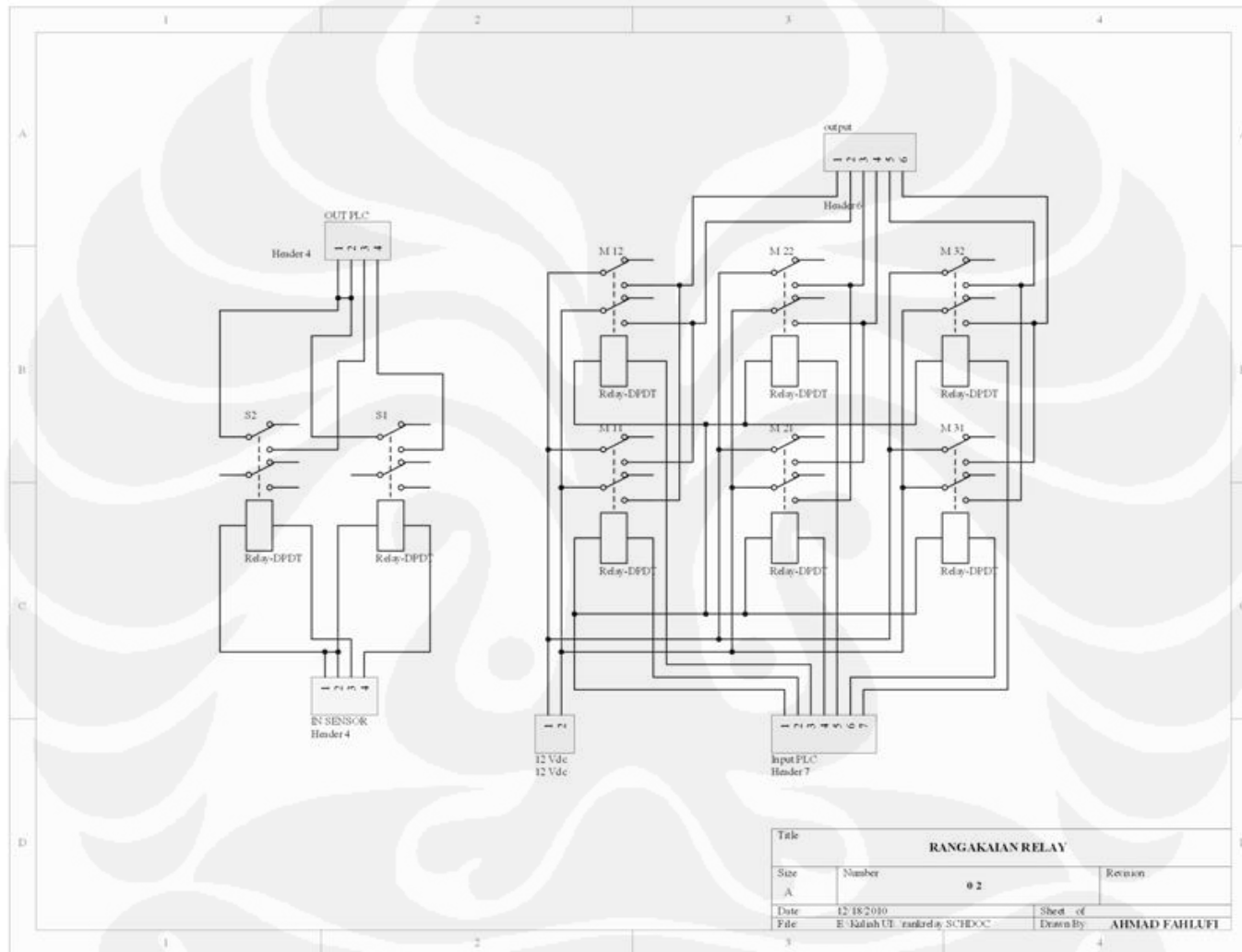


**Gambar Modul Rangkaian Relay**





**Gambar Rangkaian Modul Sensor Ketinggian Air**



**Skematik Rangkaian Relay**





## LAMPIRAN 2

**Dim X As Integer**

**Private Values() As String**

**Private NumPoints As Integer**

**Dim i As Integer**

**Dim strSql**

**' TOMBOL DELETE DATABASE AIR**

**Private Sub cmddelete\_Click()**

**X = MsgBox("Do you want to delete this data? ", vbQuestion + vbOKCancel,  
"Confirmation")**

**If X = vbOK Then**

**hapus2**

**'waterx = Text2.Text**

**'timex = Format(Now, "hh:mm")**

**'save\_Data2**

**End If**

**End Sub**

**Private Sub cmdreset\_Click()**

**CIO\_reset**

**End Sub**

**'KETINGGIAN AIR**

**Private Sub cmdsave\_Click()**

```
If Text2.Text = 0 Then  
MsgBox ("Data still empty"), vbYes, "Confirmation"  
Else  
X = MsgBox("Do you want to saving this data? ", vbQuestion + vbOKCancel,  
"Confirmation")  
If X = vbOK Then  
save2  
End If  
End If  
End Sub  
  
'HAPUS DATABASE AIR  
Private Sub hapus2()  
BukaDB  
Set rs = New ADODB.Recordset  
Sql = "DELETE * FROM WTL"  
rs.Open Sql, cn  
waterx = Text2.Text  
timex = Format(Now, "hh:mm")  
BukaDB  
save_Data2  
Adodc2.Refresh  
Chart1.Refresh  
End Sub  
  
Private Sub Darurat_Click()
```

**'Dim keluar As Integer**

**'keluar = MsgBox(OK)**

**'DoEvents**

**End**

**End Sub**

**'ALARM GATE**

**Private Sub Timer17\_Timer()**

**'ALARM GATE 3 RUSAK NAIK**

**If Shape1.Top <= 2020 Then**

**alarmon**

**rusak.Caption = "GATE 3 RISK"**

**warning.Visible = True**

**conditionString = "Gate 3 Risk"**

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

**BukaDB**

**save\_Data**

**Adodc1.Refresh**

**DataGrid1.Refresh**

**Timer17.Enabled = False**

**'ALARM GATE 2 RUSAK NAIK**

**ElseIf Shape2.Top <= 2020 Then**

**alarmon**

**rusak.Caption = "GATE 2 RISK"**

**warning.Visible = True**

**conditionString = "Gate 2 Risk"**

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

**BukaDB**

**save\_Data**

**Adodc1.Refresh**

**DataGrid1.Refresh**

**Timer17.Enabled = False**

**'ALARM GATE 1 RUSAK NAIK**

**ElseIf Shape3.Top <= 2020 Then**

**alarmon**

**rusak.Caption = "GATE 1 RISK"**

**warning.Visible = True**

**conditionString = "Gate 1 Risk"**

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

**BukaDB**

**save\_Data**

**Adodc1.Refresh**

**DataGrid1.Refresh**

**Timer17.Enabled = False**

**'ALARM GATE 3 RUSAK TURUN**



**ElseIf Shape1.Top >= 3940 Then**

**alarmon**

**rusak.Caption = "GATE 3 RISK"**

**warning.Visible = True**

**conditionString = "Gate 3 Risk"**

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

**BukaDB**

**save\_Data**

**Adodc1.Refresh**

**DataGrid1.Refresh**

**Timer17.Enabled = False**

**'ALARM GATE 2 RUSAK TURUN**

**ElseIf Shape2.Top >= 3940 Then**

**alarmon**

**rusak.Caption = "GATE 2 RISK"**

**warning.Visible = True**

**conditionString = "Gate 2 Risk"**

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

**BukaDB**

**save\_Data**

**Adodc1.Refresh**

**DataGrid1.Refresh**

**Timer17.Enabled = False**

**'ALARM GATE 1 RUSAK TURUN**

**ElseIf Shape3.Top >= 3940 Then**

**alarmon**

**rusak.Caption = "GATE 1 RISK"**

**warning.Visible = True**

**conditionString = "Gate 1 Risk"**

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

**BukaDB**

**save\_Data**

**Adodc1.Refresh**

**DataGrid1.Refresh**

**Timer17.Enabled = False**

**End If**

**End Sub**

**Private Sub Timer18\_Timer()**

**LoadData**

**' kirim data ke chart 1**

**Chart1.RowCount = NumPoints**

**Chart1.ColumnCount = 2**

**Chart1.ChartData = Values**

**Chart1.Refresh**

**waterx = Text2.Text**

**timex = Format(Now, "hh:mm")**

**BukaDB**

**save\_Data2**

**Adodc2.Refresh**

**Chart1.Refresh**

**End Sub**

**Private Sub up\_Click()**

**Text2.Text = X + 1**

**CIO\_naik**

**End Sub**

**Private Sub down\_Click()**

**Text2.Text = X - 1**

**CIO\_turun**

**End Sub**

**Private Sub cmdclose\_Click()**

**Dim out As Integer**

**out = MsgBox("Do you want to save this project", vbQuestion +  
vbYesNoCancel, "Confirmation")**

**If out = vbNo Then**

**hapus**

**DoEvents**

**End**

**ElseIf out = vbYes Then**

**save**

**DoEvents**

```
End
End If
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
'SAVE DATABASE KONDISI JIKA KELUAR
Private Sub save()
BukaDB
Set rs = New ADODB.Recordset
Sql = "SAVE * FROM data"
Adodc1.Refresh
DataGrid1.Refresh
End Sub
'SETTING KETINGGIAN AIR
Private Sub waterset_Click()
up.Enabled = True
down.Enabled = True
finish.Visible = True
waterset.Visible = False
End Sub
```

**'BUKA SETTING AIR**

**Private Sub finish\_Click()**

**up.Enabled = False**

**down.Enabled = False**

**finish.Visible = False**

**waterset.Visible = True**

**End Sub**

**Private Sub Form\_Load()**

**'INISIALISASI DTABASE**

**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**

**Adodc2.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & "\Data.mdb"**

**Adodc2.RecordSource = "WTL"**

**Adodc2.Refresh**

**LoadData**

**' kirim data ke chart 1**

**Chart1.RowCount = NumPoints**

**Chart1.ColumnCount = 2**

**Chart1.ChartData = Values**

**Chart1.Refresh**

```
'sistem start dengan mode auto
cmdopen1.Enabled = False
cmdopen2.Enabled = False
cmdopen3.Enabled = False
cmdclose1.Enabled = False
cmdclose2.Enabled = False
cmdclose3.Enabled = False
SYSMAC_C1.CIO(31) = 1
SYSMAC_C1.CIO(31) = 0
End Sub
Private Sub LoadData()
'Buka database
BukaDB
'Lihat jumlah datanya
strSql = "SELECT Time, Height FROM WTL"
rs.Open strSql, cn
rs.MoveLast
NumPoints = rs.RecordCount
ReDim Values(1 To NumPoints, 1 To 2)
'Load data.
rs.MoveFirst
For i = 1 To NumPoints
Values(i, 1) = rs("Time")
Values(i, 2) = rs("Height")
```

**rs.MoveNext**

**Next i**

**'Tutup koneksi den database**

**Set rs = Nothing**

**cn.Errors.Clear**

**cn.Close**

**End Sub**

**Private Sub cmdalarm\_Click()**

**CIO\_alarm**

**End Sub**

**Private Sub cmdclose1\_Click()**

**CIO\_close1**

**End Sub**

**Private Sub cmdclose2\_Click()**

**CIO\_close2**

**End Sub**

**Private Sub cmdclose3\_Click()**

**CIO\_close3**

**End Sub**

**Private Sub cmdopen1\_Click()**

**CIO\_open1**

**End Sub**

**Private Sub cmdopen2\_Click()**

**CIO\_open2**

**End Sub**

**Private Sub cmdopen3\_Click()**

**CIO\_open3**

**End Sub**

**Private Sub cmdautoman\_Click()**

**If cmdautoman.Tag = 0 Then**

**CIO\_manual**

**cmdautoman.Tag = 1**

**cmdautoman.Picture = cmdautoman.DownPicture**

**lbldisplay.Caption = "MANUAL SYSTEM"**

**cmdopen1.Enabled = True**

**cmdopen2.Enabled = True**

**cmdopen3.Enabled = True**

**cmdclose1.Enabled = True**

**cmdclose2.Enabled = True**

**cmdclose3.Enabled = True**

**waterset.Enabled = True**



```
Else
CIO_auto
cmdautoman.Tag = 0
cmdautoman.Picture = Picture2
lbldisplay.Caption = "AUTO SYSTEM"
cmdopen1.Enabled = False
cmdopen2.Enabled = False
cmdopen3.Enabled = False
cmdclose1.Enabled = False
cmdclose2.Enabled = False
cmdclose3.Enabled = False
up.Enabled = False
down.Enabled = False
finish.Visible = False
waterset.Visible = True
waterset.Enabled = False
End If

End Sub

Private Sub CIO_auto()
'Auto System
SYSMAC_C1.CIO(31) = 1
SYSMAC_C1.CIO(31) = 0
End Sub

Private Sub CIO_manual()
```

```
'Manual System
SYSMAC_C1.CIO(30) = 1
SYSMAC_C1.CIO(30) = 0
End Sub
Private Sub CIO_open1()
'Pintu 1 Buka
SYSMAC_C1.CIO(40) = 1
SYSMAC_C1.CIO(40) = 0
Timer11.Enabled = True
End Sub
Private Sub CIO_close1()
'Pintu 1 Turun
SYSMAC_C1.CIO(41) = 1
SYSMAC_C1.CIO(41) = 0
Timer12.Enabled = True
End Sub
Private Sub CIO_open2()
'Pintu 2 Buka
SYSMAC_C1.CIO(42) = 1
SYSMAC_C1.CIO(42) = 0
Timer13.Enabled = True
End Sub
Private Sub CIO_close2()
'Pintu 2 Turun
SYSMAC_C1.CIO(43) = 1
SYSMAC_C1.CIO(43) = 0
```

**Timer14.Enabled = True**

**End Sub**

**Private Sub CIO\_open3()**

**'Pintu 3 Buka**

**SYSMAC\_C1.CIO(44) = 1**

**SYSMAC\_C1.CIO(44) = 0**

**Timer15.Enabled = True**

**End Sub**

**Private Sub CIO\_close3()**

**'Pintu 3 Turun**

**SYSMAC\_C1.CIO(45) = 1**

**SYSMAC\_C1.CIO(45) = 0**

**Timer16.Enabled = True**

**End Sub**

**Private Sub CIO\_reset()**

**'Reset**

**SYSMAC\_C1.CIO(32) = 1**

**SYSMAC\_C1.CIO(32) = 0**

**End Sub**

**Private Sub CIO\_alarm()**

**'Alarm off**

**SYSMAC\_C1.CIO(23) = 1**

**SYSMAC\_C1.CIO(23) = 0**

**End Sub**

**Private Sub alarmon()**

**'Alarm on**

**SYSMAC\_C1.CIO(34) = 1**

**SYSMAC\_C1.CIO(34) = 0**

**End Sub**

**Private Sub CIO\_naik()**

**'Sensor Naik**

**SYSMAC\_C1.CIO(24) = 1**

**SYSMAC\_C1.CIO(24) = 0**

**End Sub**

**Private Sub CIO\_turun()**

**'Sensor Turun**

**SYSMAC\_C1.CIO(25) = 1**

**SYSMAC\_C1.CIO(25) = 0**

**End Sub**

**Private Sub cmdctv\_Click()**

**frmwebcam.Show**

**End Sub**

```
'Private Sub cmdtdown_Click()
```

```
'Text2.Text = Text1.Text
```

```
'If X > 0 Then
```

```
'X = X - 1
```

```
'End If
```

```
'End Sub
```

```
'Private Sub cmdup_Click()
```

```
'X = X + 1
```

```
'Text1.Text = X
```

```
'End Sub
```

```
Private Sub Timer1_Timer()
```

```
jam.Caption = Format(Now, "hh:mm:ss")
```

```
End Sub
```

```
Private Sub tanki()
```

```
X = Text2.Text
```

```
belakang.Height = 3615 - (X * 125)
```

```
End Sub
```

```
Private Sub Timer10_Timer()
```

```
tanki
```

```
End Sub
```

```
Private Sub Timer2_Timer()
```

```
Text2.Text = SYSMAC_C1.DM(0)

End Sub

Private Sub Timer11_Timer()

conditionString = "Gate 1 Open"

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

BukaDB

save_Data

Adodc1.Refresh

DataGrid1.Refresh

Timer11.Enabled = False

End Sub

Private Sub Timer12_Timer()

conditionString = "Gate 1 Close"

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

BukaDB

save_Data

Adodc1.Refresh

DataGrid1.Refresh

Timer12.Enabled = False

End Sub

Private Sub Timer13_Timer()

conditionString = "Gate 2 Open"

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

BukaDB
```

```
save_Data
Adodc1.Refresh
DataGrid1.Refresh
Timer13.Enabled = False
End Sub
Private Sub Timer14_Timer()
conditionString = "Gate 2 Close"
waktuString = Format(Date, "dd/mm/yyyy") & " " & Format(Now,
"hh:mm")
BukaDB
save_Data
Adodc1.Refresh
DataGrid1.Refresh
Timer14.Enabled = False
End Sub
Private Sub Timer15_Timer()
conditionString = "Gate 3 Open"
waktuString = Format(Date, "dd/mm/yyyy") & " " & Format(Now,
"hh:mm")
BukaDB
save_Data
Adodc1.Refresh
DataGrid1.Refresh
Timer15.Enabled = False
End Sub
Private Sub Timer16_Timer()
conditionString = "Gate 3 Close"
```

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

```
BukaDB
```

```
save_Data
```

```
Adodc1.Refresh
```

```
DataGrid1.Refresh
```

```
Timer16.Enabled = False
```

```
End Sub
```

```
Private Sub Timer3_Timer()
```

```
prosedur_readstatus
```

```
End Sub
```

```
Private Sub prosedur_readstatus()
```

```
'STATUS AIR NAIK
```

```
'If SYSMAC_C1.DM(6) = 1 Then
```

```
'Timer18.Enabled = True
```

```
'End If
```

```
'STATUS AIR TURUN
```

```
'If SYSMAC_C1.DM(6) = 2 Then
```

```
'Timer18.Enabled = True
```

```
'End If
```

```
'STATUS PINTU 1 NAIK
```

```
If SYSMAC_C1.DM(1) = 1 Then
```

```
Timer8.Enabled = True
```

```
Label9.Caption = "OPENED"
```



**Shape6.BackColor = &HFF00&**

**Timer11.Enabled = True**

**End If**

**'STATUS PINTU 1 TURUN**

**If SYSMAC\_C1.DM(1) = 2 Then**

**Timer9.Enabled = True**

**Label9.Caption = "CLOSED"**

**Shape7.BackColor = &HFF&**

**Timer12.Enabled = True**

**End If**

**'STATUS PINTU 2 NAIK**

**If SYSMAC\_C1.DM(2) = 1 Then**

**Timer6.Enabled = True**

**Label8.Caption = "OPENED"**

**Shape9.BackColor = &HFF00&**

**Timer13.Enabled = True**

**End If**

**'STATUS PINTU 2 TURUN**

**If SYSMAC\_C1.DM(2) = 2 Then**

**Timer7.Enabled = True**

**Label8.Caption = "CLOSED"**

**Shape10.BackColor = &HFF&**

**Timer14.Enabled = True**

**End If**

**'Status PINTU 3 NAIK**

**If SYSMAC\_C1.DM(3) = 1 Then**

**Timer4.Enabled = True**

**Label4.Caption = "OPENED"**

**Shape11.BackColor = &HFF00&**

**Timer15.Enabled = True**

**End If**

**'STATUS PINTU 3 TURUN**

**If SYSMAC\_C1.DM(3) = 2 Then**

**Timer5.Enabled = True**

**Label4.Caption = "CLOSED"**

**Shape12.BackColor = &HFF&**

**Timer16.Enabled = True**

**End If**

**'STATUS ALARM**

**If SYSMAC\_C1.DM(4) = 0 Then**

**Picture4.Visible = True**

**Picture3.Visible = False**

**ElseIf SYSMAC\_C1.DM(4) = 1 Then**

**Picture4.Visible = False**

**Picture3.Visible = True**

**End If**

**'STATUS ALARM STOP**

**If SYSMAC\_C1.DM(5) = 1 Then**

**conditionString = "Alarm 0N"**

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

**BukaDB**

**save\_Data**

**Adodc1.Refresh**

**DataGrid1.Refresh**

**ElseIf SYSMAC\_C1.DM(5) = 2 Then**

**conditionString = "Alarm 0FF"**

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

**BukaDB**

**save\_Data**

**Adodc1.Refresh**

**DataGrid1.Refresh**

**End If**

**End Sub**

**Private Sub Timer4\_Timer()**

**Shape1.Top = Shape1.Top - 10**

**If Shape1.Top = 2040 Then**

**Timer4.Enabled = False**

**Shape11.BackColor = &H4000&**

**End If**

**End Sub**

**Private Sub Timer5\_Timer()**

**Shape1.Top = Shape1.Top + 10**

**If Shape1.Top = 3930 Then**

**Timer5.Enabled = False**

**Shape12.BackColor = &H40&**

**Label4.Caption = "STAND BY"**

**End If**

**End Sub**

**Private Sub Timer6\_Timer()**

**Shape2.Top = Shape2.Top - 10**

**If Shape2.Top = 2040 Then**

**Shape9.BackColor = &H4000&**

**Timer6.Enabled = False**

**End If**

**End Sub**

**Private Sub Timer7\_Timer()**

**Shape2.Top = Shape2.Top + 10**

**If Shape2.Top = 3930 Then**

**Timer7.Enabled = False**

```
Shape10.BackColor = &H40&  
Label8.Caption = "STAND BY"  
End If  
End Sub
```

```
Private Sub Timer8_Timer()  
Shape3.Top = Shape3.Top - 10  
If Shape3.Top = 2040 Then  
Timer8.Enabled = False  
Shape6.BackColor = &H4000&  
End If  
End Sub
```

```
Private Sub Timer9_Timer()  
Shape3.Top = Shape3.Top + 10  
If Shape3.Top = 3930 Then  
Shape7.BackColor = &H40&  
Timer9.Enabled = False  
Label9.Caption = "STAND BY"  
End If  
End Sub
```

```
Private Sub save2()  
On Error GoTo batal  
Dim AppExcel As Excel.Application  
Dim ExcelWBk As Excel.Workbook  
Dim ExcelWS As Excel.Worksheet
```

**Dim ColField As Collection**

**Dim jmlTabel As Integer**

**Dim jmlField As Integer**

**Screen.MousePointer = vbHourglass**

**BukaDB**

**Dim rs As ADODB.Recordset**

**Set rs = New ADODB.Recordset**

**rs.CursorLocation = adUseClient**

**rs.LockType = adLockPessimistic**

**rs.Open "WTL", strcn**

**cmd.Filter = "Excel Files (\*.xlsx)"**

**cmd.ShowSave**

**Set AppExcel = New Excel.Application**

**Set ExcelWBk = AppExcel.Workbooks.Add**

**Set ExcelWS = ExcelWBk.Worksheets.Add**

**ExcelWS.Name = "Data"**

**Set ColField = New Collection**

**For jmlField = 0 To rs.Fields.Count - 1**

**ExcelWS.Cells(3, jmlField + 1) = rs.Fields(jmlField).Name**

**ColField.Add rs.Fields(jmlField).Name**

**DoEvents**

**Next jmlField**

**Dim Pos As Long**

```
Pos = 3
If Not rs.EOF Then
rs.MoveFirst
While Not rs.EOF
For jmlField = 1 To ColField.Count
ExcelWS.Cells(Pos, jmlField) = rs(ColField(jmlField))
Next jmlField
Pos = Pos + 1
rs.MoveNext
DoEvents
Wend
End If
With ExcelWS
.Cells(1, 2).Value = "DATA WATER LEVEL"
.Cells(1, 2).Font.Bold = True:
.Cells(1, 2).Font.Underline = True:
.Cells(2, 1).Value = "No"
.Cells(2, 1).Font.Bold = True:

.Cells(2, 2).Value = "Water Level (cm)"
.Cells(2, 2).Font.Bold = True:

.Cells(2, 3).Value = "Time"
.Cells(2, 3).Font.Bold = True:

End With
```

**last:**

**ExcelWBk.SaveAs comd.FileName**

**ExcelWBk.Saved = True**

**AppExcel.Quit**

**Set AppExcel = Nothing**

**Set ExcelWS = Nothing**

**Set rs = Nothing**

**Screen.MousePointer = vbDefault**

**MsgBox "Save Data Complete"**

**Exit Sub**

**batal:**

**MsgBox "Cancel saved"**

**On Error Resume Next**

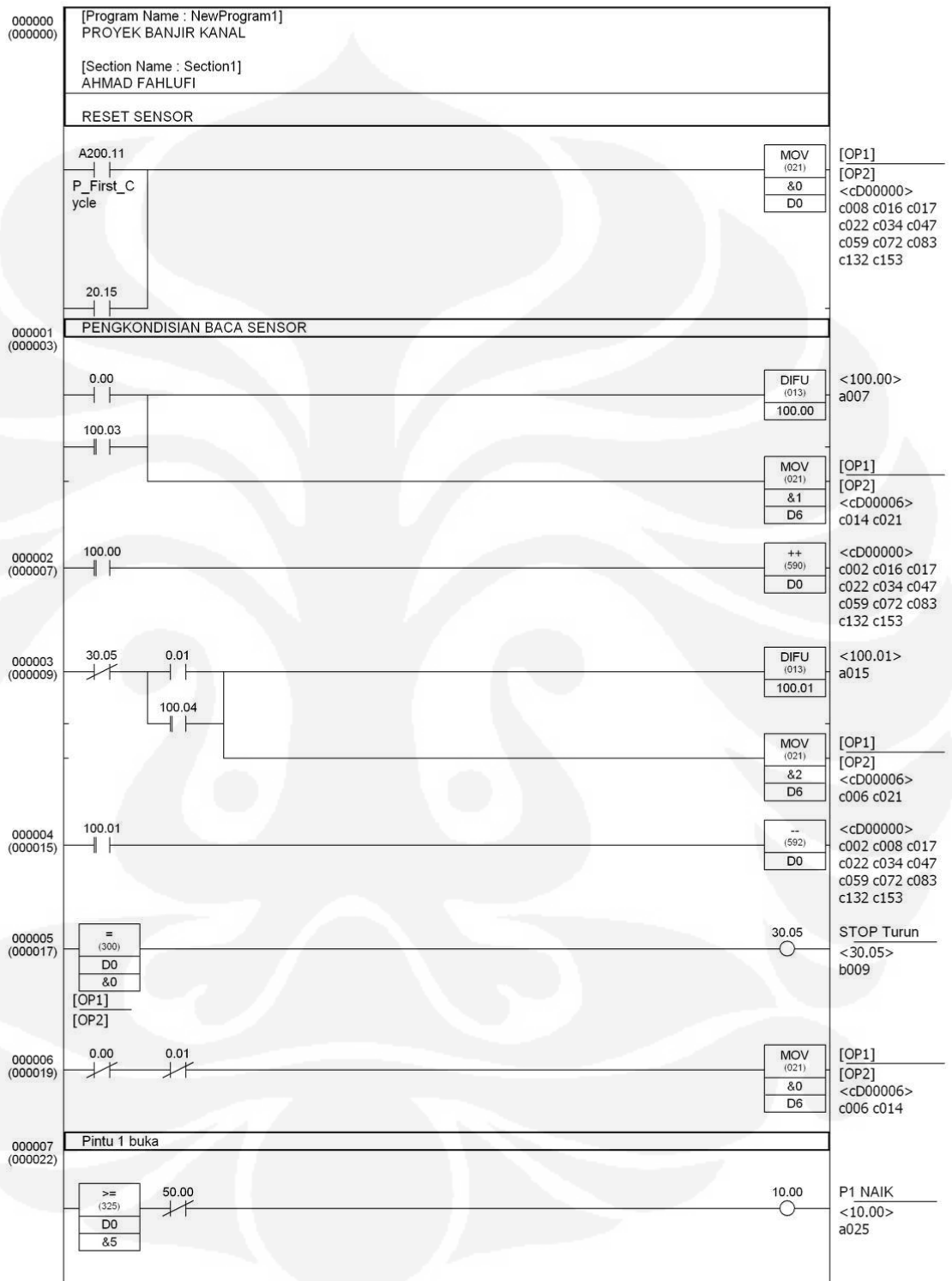
**Screen.MousePointer = vbDefault**

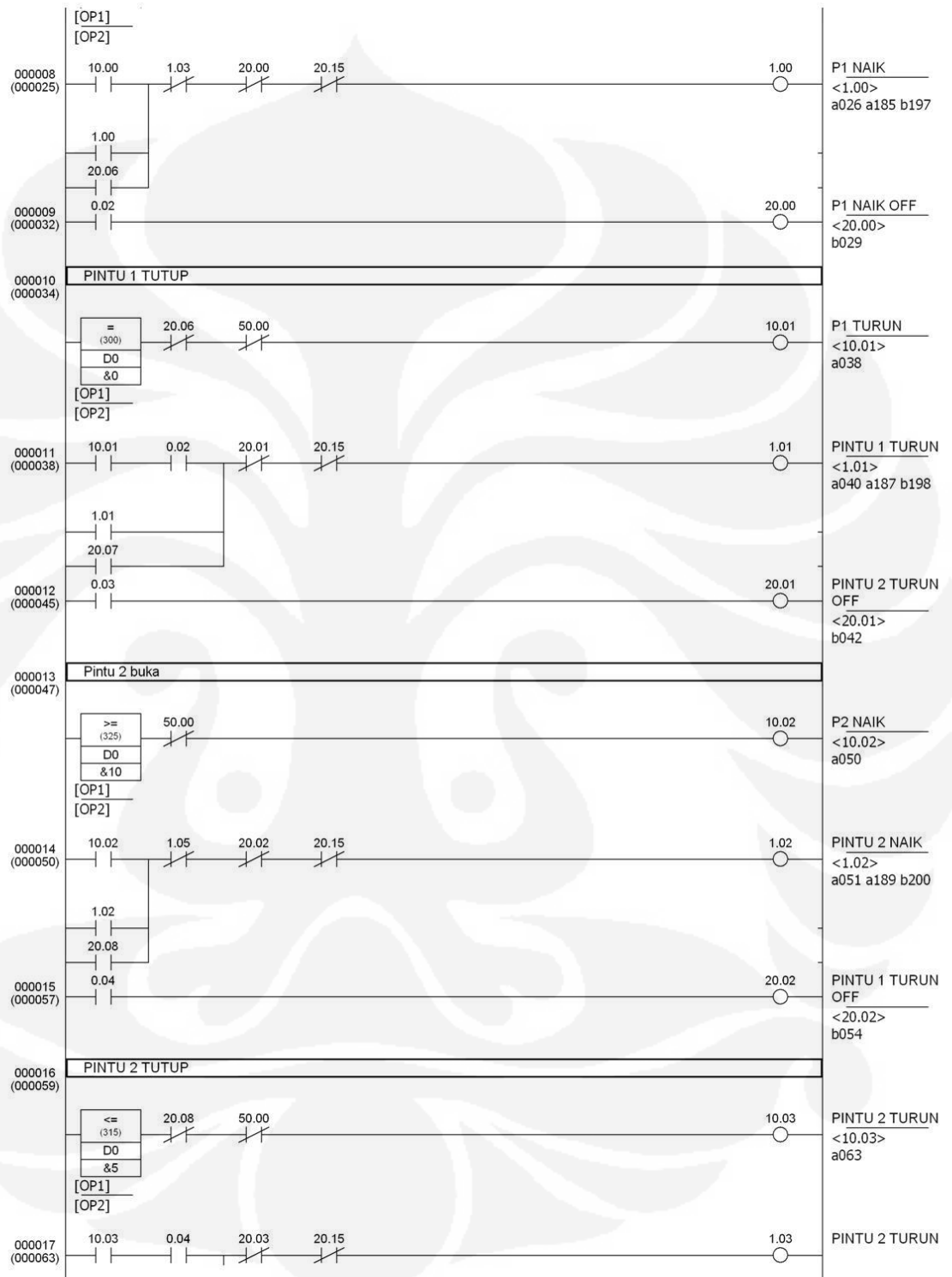
**End Sub**

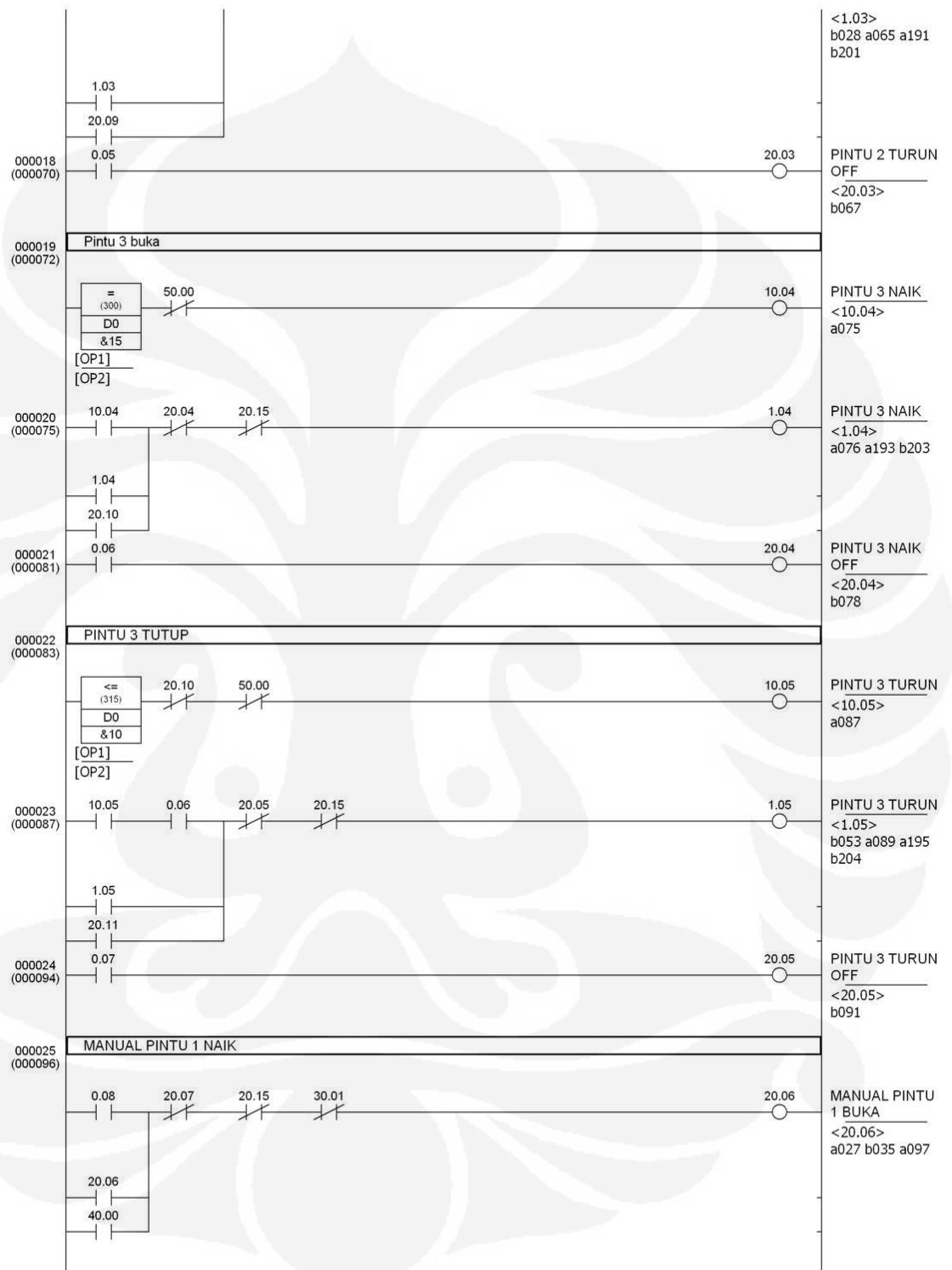


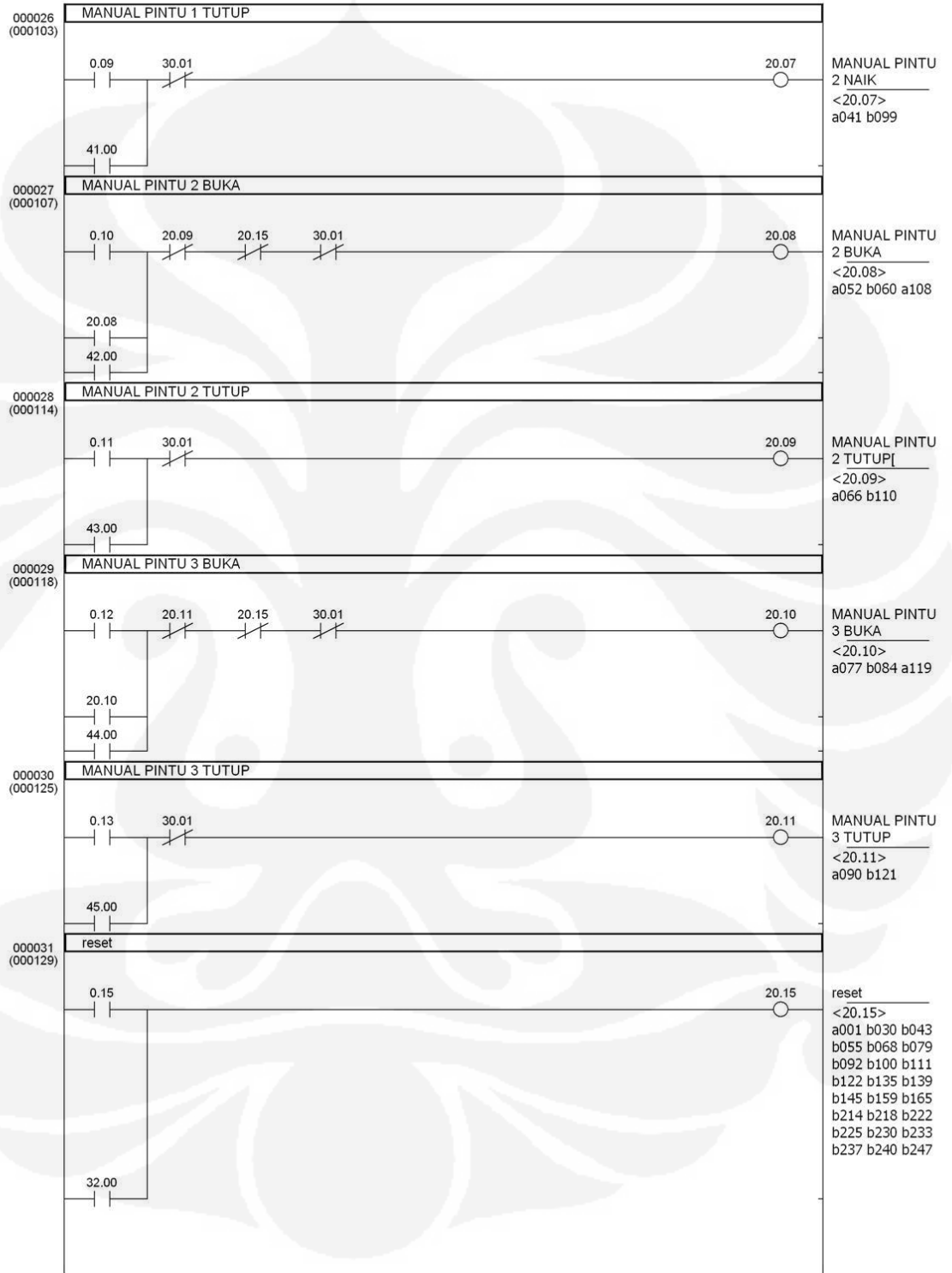


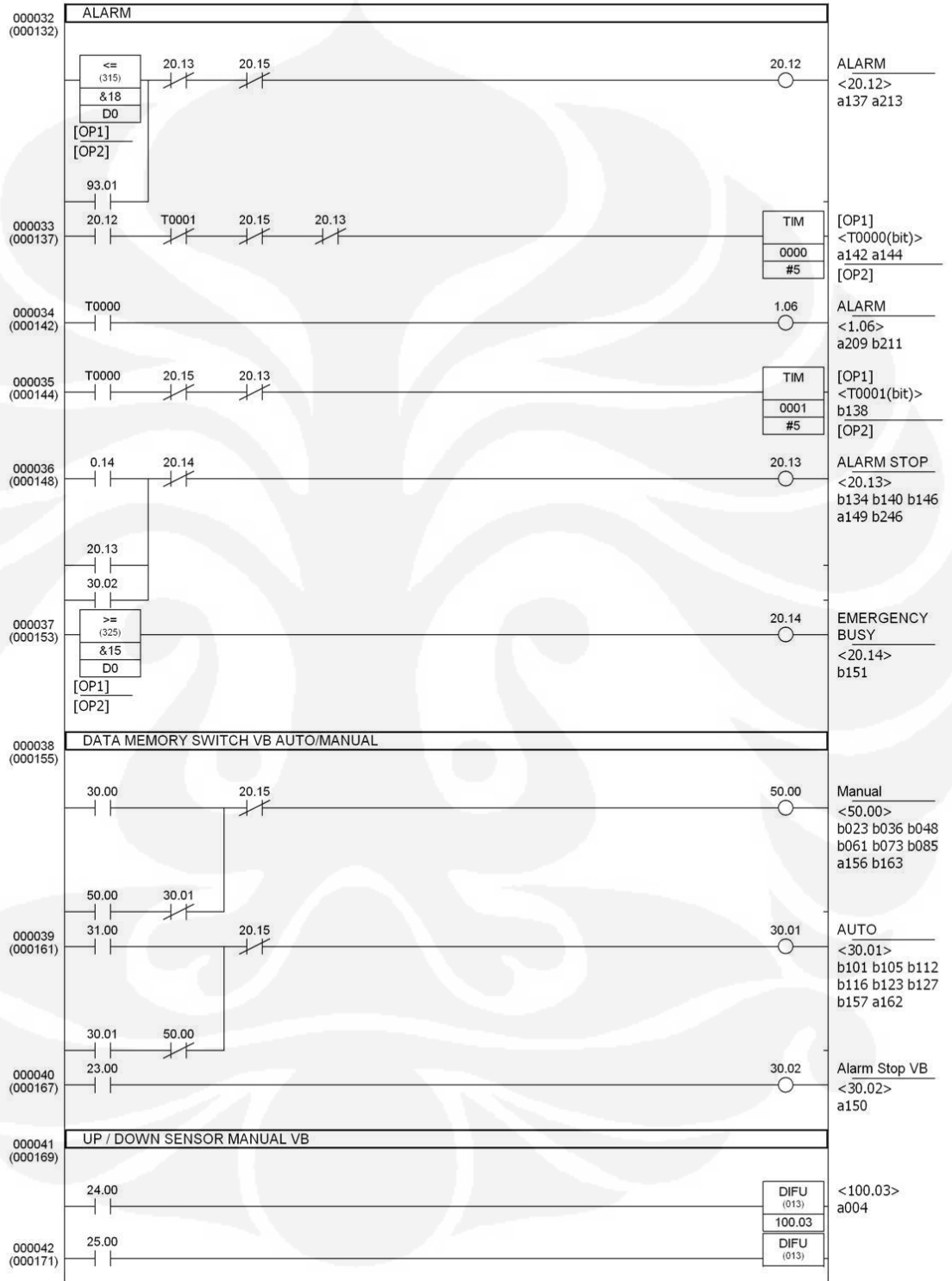
## LAMPIRAN 3

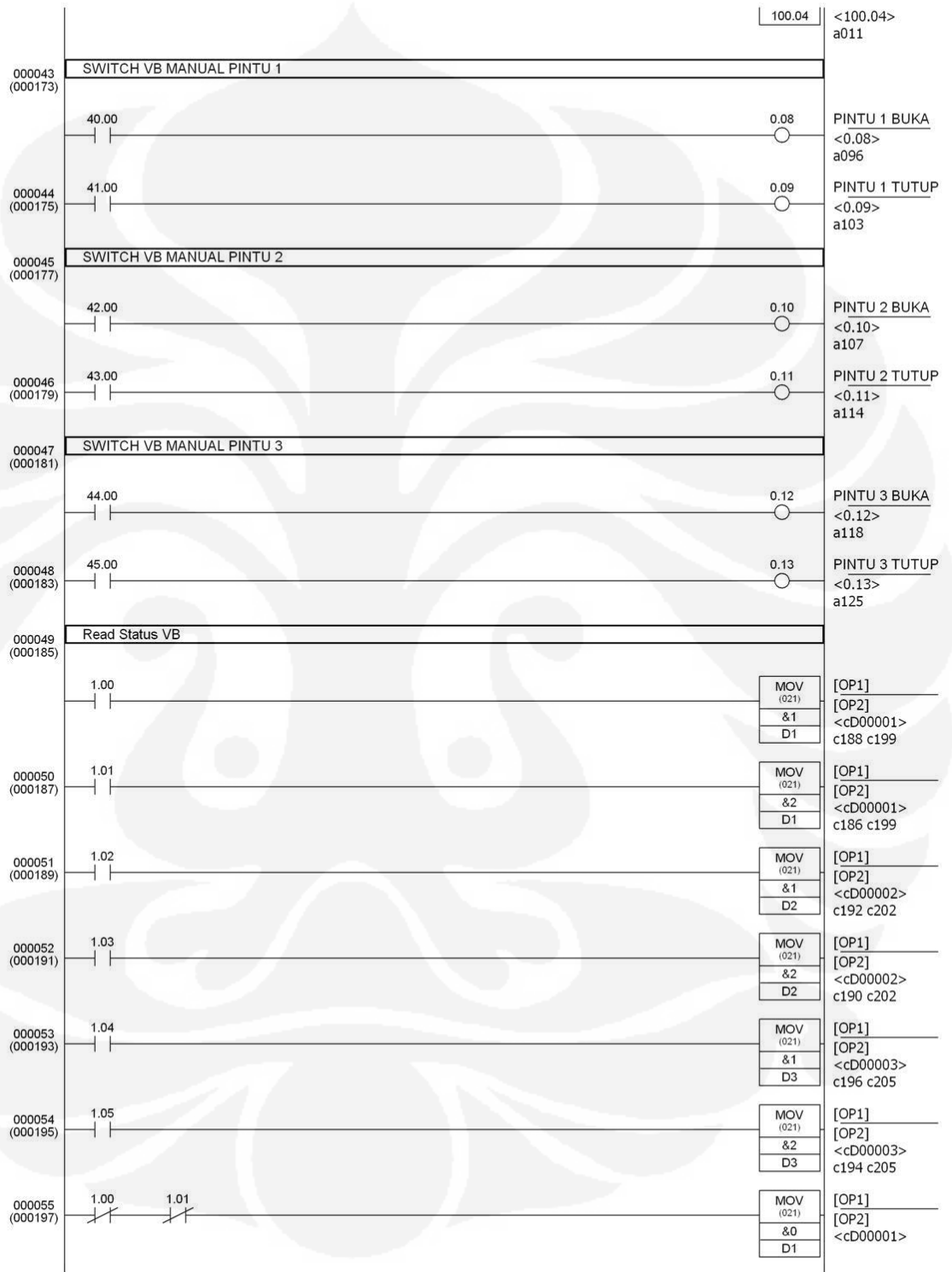


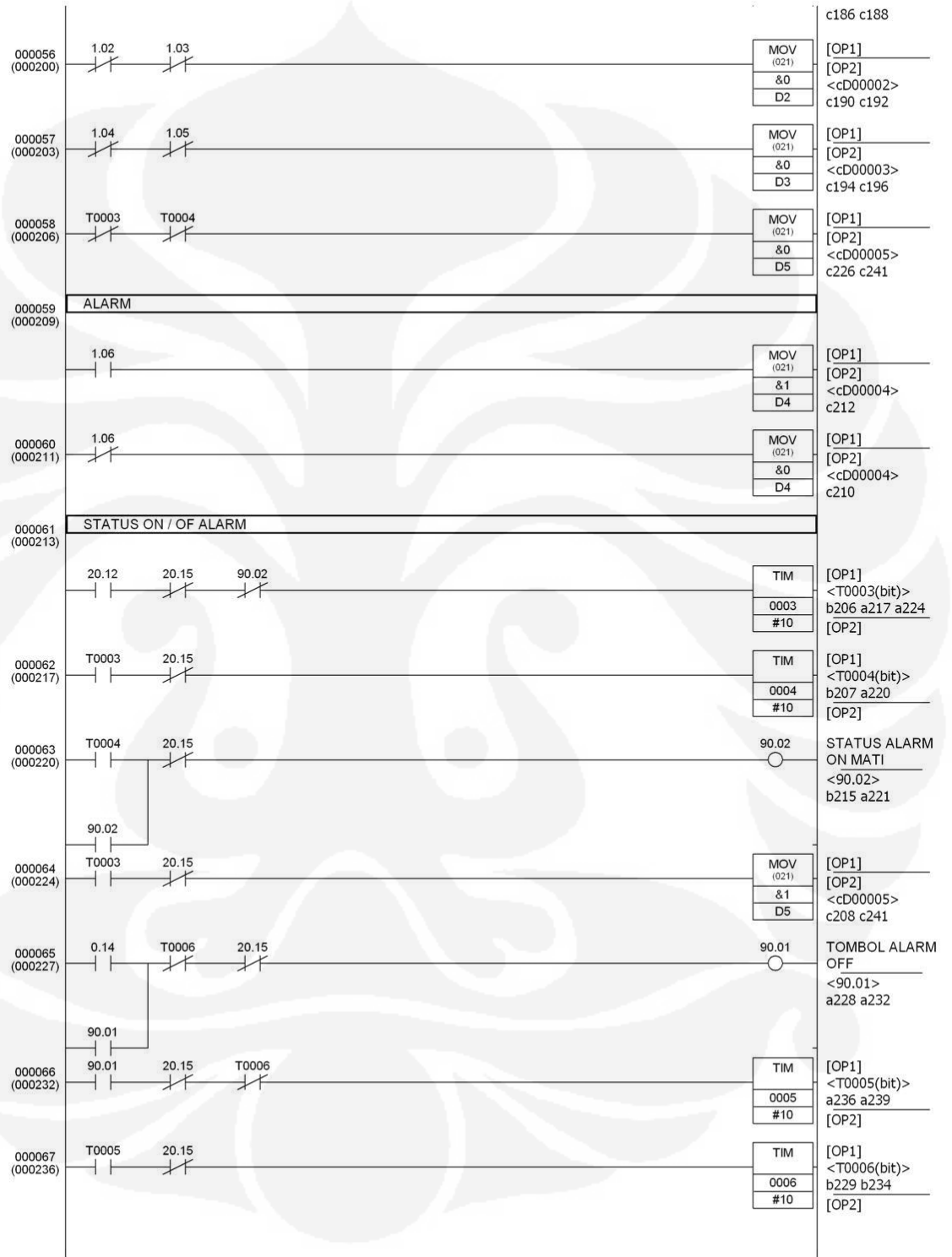




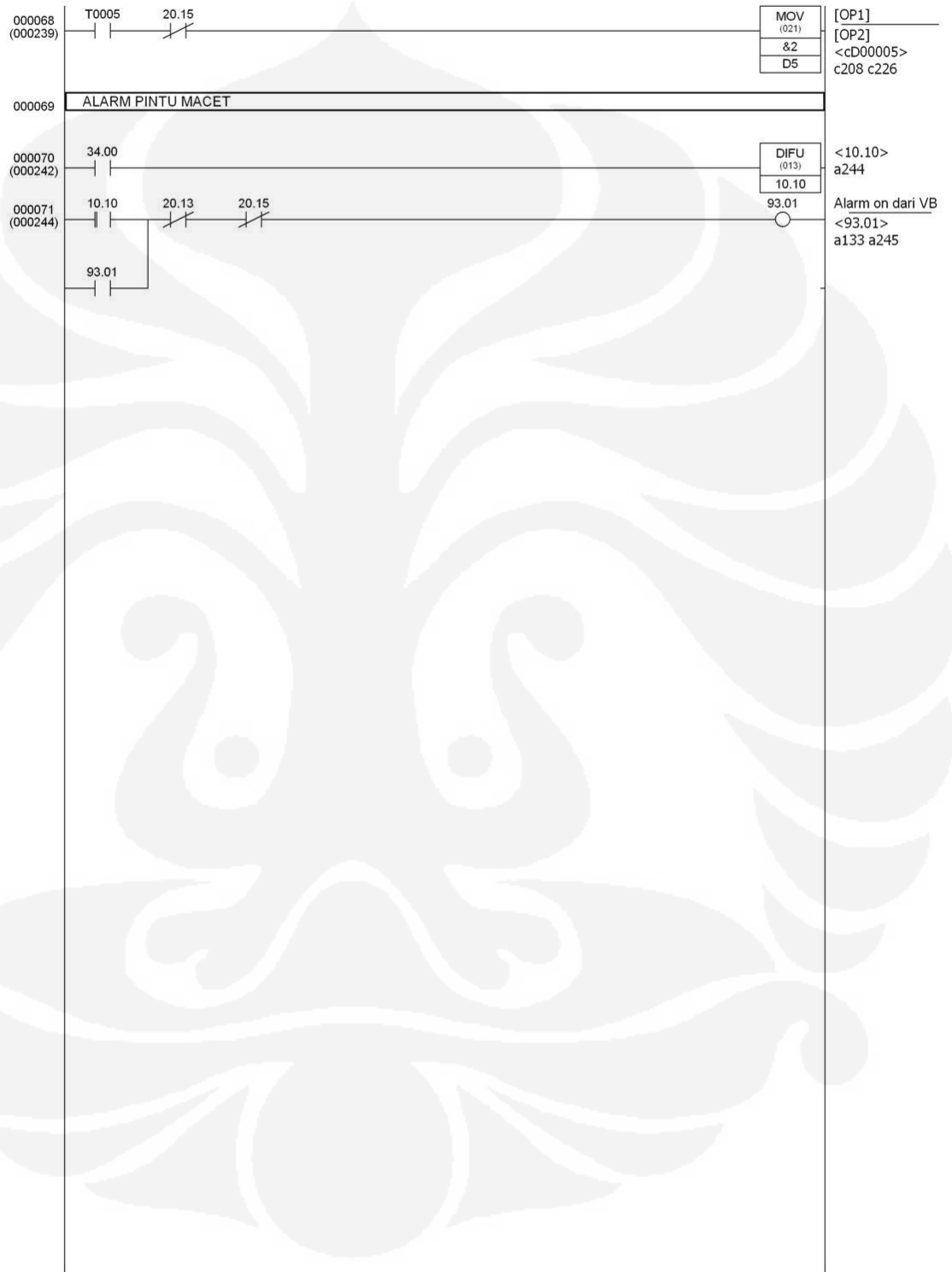














## LAMPIRAN 4

# SYSMAC CJ-series Input Units

## CJ1W-ID/IA

PLC CJ1W-ID/IA 08 E 4.4

### 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 $\mu$ s, OFF Response Time: 90 $\mu$ 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

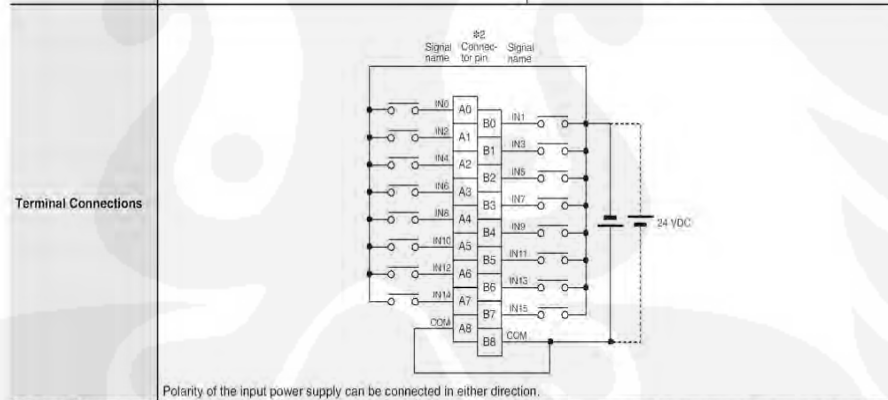
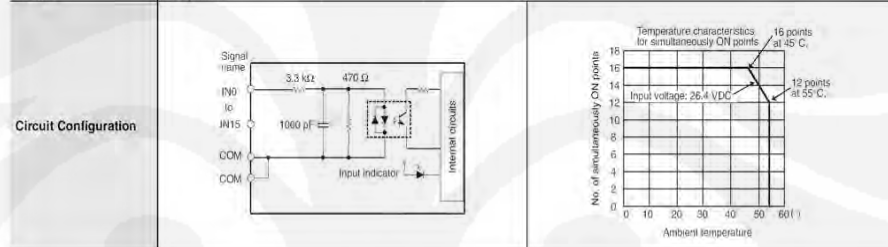
Unit type	Product name	Specifications					Current consumption (A)		Model	Standards	
		I/O points	Input voltage and current	Commons	External connection	No. of words allocated	5 V	24 V			
CJ1 Basic I/O Units	DC Input Units	8 inputs	12 to 24 VDC, 10 mA	Independent contacts	Removable terminal block	1 word	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	—	<b>NEW</b> 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	—	<b>NEW</b> CJ1W-ID233		
		64 inputs	24 VDC, 4.1 mA	16 points, 1 common	Fujitsu connector	4 words	0.09	—	CJ1W-ID261	N, L, CE	
		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 kΩ
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 MΩ 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 μs maximum and OFF response time will be 400 μs 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

CsM\_CJ1W-OUTPUT\_Ds\_E\_U\_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 $\mu$ s, OFF Response Time: 80 $\mu$ 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




OMRON

1



## Ordering Information

## Output Units

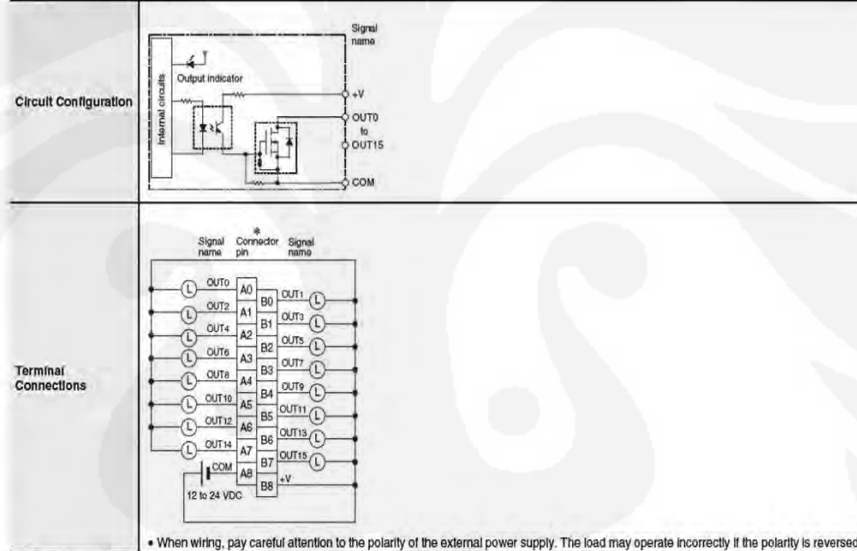
Unit type	Product name	Specifications					No. of words allocated	Current consumption (A)		Model	Standards	
		Output type	I/O points	Maximum switching capacity	Commons	External connection		5 V	24 V			
CJ1 Basic I/O Units	 Relay Contact Output Units	-	8 outputs	250 VAC/24 VDC, 2 A	Independent 1 contacts	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	-	<sup>NEW</sup> 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		UC1, N, L, CE
		Sinking	32 outputs (High speed)	24 VDC, 0.5 A	16 points, 1 common	MIL connector	2 words	0.22	-	<sup>NEW</sup> 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)**

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





## 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).

- 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.



CJ1W-PA205C



CJ1W-PA202

### 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.

