



UNIVERSITAS INDONESIA

**PERANCANGAN SISTEM ANTARMUKA BERBASIS HMI
UNTUK APLIKASI MESIN ETCHING PCB**

SKRIPSI

JASON MIDIAN FD

07 06 19 9483

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DESEMBER, 2009**



UNIVERSITAS INDONESIA

**PERANCANGAN SISTEM ANTARMUKA BERBASIS HMI
UNTUK APLIKASI MESIN ETCHING PCB**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

JASON MIDIAN FD

07 06 19 9483

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

DESEMBER, 2009

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Jason Midian FD

NPM : 0706199483

Tanda Tangan :

Tanggal : 28 Desember 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Jason Midian FD
NPM : 0706199483
Program Studi : Teknik Elektro
Judul Skripsi : Perancangan Sistem Antarmuka Berbasis HMI
Untuk Aplikasi Mesin Etching PCB

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

DEWAN PENGUJI

Pembimbing : Dr. Abdul Halim M.Eng ()
Penguji : Dr. Ir. Feri Yusivar, M.Eng ()
Penguji : Dr. Ridwan Gunawan MT ()

Ditetapkan di : Depok

Tanggal : 28 Desember 2009

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

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

Nama : Jason Midian FD
NPM : 0706199483
Program Studi : Kontrol
Departemen : 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 :

**PERANCANGAN SISTEM ANTARMUKA BERBASIS HMI
UNTUK APLIKASI MESN ETCHING PCB**

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 mempublikasikan tugas akhir 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 : 28 Desember 2009

Yang menyatakan

(Jason Midian FD)

KATA PENGANTAR

Dengan memanjatkan puja dan puji syukur kepada Bapa di Surga serta dengan kuasa Tuhan Yesus Kristus dan Roh Kudus yang atas berkah dan kasihNya penulis berhasil menyelesaikan skripsi yang berjudul "Perancangan Sistem Antarmuka Berbasis HMI Untuk Aplikasi Mesin Etching PCB" dengan baik. Skripsi ini dibuat dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) di Universitas Indonesia pada jurusan Teknik program studi Teknik Elektro.

Penulis berharap semoga skripsi ini dapat berguna bagi penulis dan pembaca. Penulisan skripsi ini sejak dari persiapan sampai penulisan, penulis mendapat bimbingan, dukungan semangat dan moril dari berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih yang ditujukan kepada:

1. Dr. Abdul Halim M.Eng, selaku dosen Pembimbing.
2. Rekan Robin yang telah membantu penulis dalam berdiskusi tentang listrik.
3. Rekan-rekan ekstensi 2007 yang telah banyak membantu penulis dalam menyelesaikan skripsi ini.
4. Terakhir dan terutama adalah kepada seluruh keluarga, karena dengan bimbingan, motivasi serta doa dari mereka, penulis dapat menyelesaikan skripsi ini.

Penulis menyadari bahwa dalam skripsi ini masih banyak kekurangan, untuk itu penulis mengharapkan kritik dan saran yang membangun agar skripsi ini dapat menjadi lebih baik. Akhir kata penulis berharap skripsi ini berguna bagi penulis dan yang membacanya.

Depok, Desember 2009

Penulis

ABSTRAK

Nama : Jason Midian FD
Program Studi : Teknik Elektro
Judul : Perancangan Sistem Antarmuka Berbasis HMI
Untuk Aplikasi Mesin Etching PCB

Human Machine Interface (HMI) sudah umum dipergunakan di dunia industri. Fasilitas-fasilitas yang ditampilkan oleh HMI dapat mempermudah pekerja dalam melakukan proses pekerjaannya seperti antara lain program data-entry, program permintaan keterangan, pembuatan laporan dan dokumen. Dalam skripsi ini, keunggulan yang disediakan oleh HMI dimanfaatkan untuk sistem antarmuka bagi aplikasi mesin etching PCB.

Untuk membuat sistem antar muka berbasis HMI pada aplikasi mesin etching PCB, perlu dirancang alat-alat yang mendukung penuh sistem ini. Sistem ini terdiri dari komponen hardware dan software. Komponen hardware utama meliputi PLC (Programmable Logic Controller), Microcontroller AT89s51, PC, prototipe mesin etching PCB. Sedangkan software meliputi program-program yang dibangun dengan software Ladder Syswin 3.4, Assembler51 dan Visual Basic 6.

Sistem antar muka yang dirancang dapat menggerakkan 2 buah mesin etching PCB secara bersamaan. Sistem yang dibangun ini dapat menekan biaya peralatan dengan signifikan karena dipergunakannya microcontroller yang menggantikan peran Device Converter (Penghubung PLC dan PC). Sistem yang dirancang dapat dikembangkan lagi agar memiliki fungsi komunikasi via jaringan internet dan untuk jumlah mesin yang lebih banyak.

Kata kunci:
PLC, HMI, Microcontroller AT89s51, Visual Basic 6

ABSTRACT

Name : Jason Midian FD
Study Program: Electrical Engineering
Title : Design Of Interfacing System Based On HMI
For PCB Etching Machine Application

Human Machine Interface (HMI) is commonly used in industry. Facilities presented by HMI can ease workers in their jobs such as data-entry program, program inquiries, preparing reports and documents. In this thesis, the benefits provided by the HMI interface systems has been used for the application PCB etching machine.

To create a system based on HMI interface for the application PCB etching machine, some tools have been required to fully support this system. This system consists of hardware and software components. The main hardware components include the PLC (Programmable Logic Controller), Microcontroller AT89s51, PC, prototype PCB etching machine. While the software includes programs that are built with software Ladder Syswin 3.4, Assembler51 and Visual Basic 6.

Interfacing system developed is used to move 2 pieces PCB etching machines simultaneously. The system can reduce significantly the cost of equipment because of the use of microcontroller that replaces the role of the Device Converter (PLC and PC Connector). The system can be developed to have a communication function via the Internet network and for more than 2 machines.

Key words:
PLC, HMI, Microcontroller AT89s51, Visual Basic 6

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	v
KATA PENGANTAR	vi
ABSTRAK	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
DAFTAR SINGKATAN	xv
DAFTAR ISTILAH	xvi
1. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Penulisan.....	2
1.4 Batasan Permasalahan.....	3
1.5 Sistematika Penulisan	3
2. LANDASAN TEORI.....	4
2.1 PLC	4
2.1.1 Sourcing Dan Sinking	5
2.1.2 Unit CPU.....	5
2.1.3 Unit Catu Daya (Power Supply)	6
2.1.4 Ladder Syswin 3.4.....	6

2.2 Microcontroller AT89s51.....	7
2.2.1 TMOD (Timer Mode)	8
2.2.2 SCON (Serial Control).....	10
2.3 Pengaksesan MSComm Pada Visual Basic 6.....	13
2.3.1 Properti MSComm	13
2.3.2 Even Pada MSComm	14
3. PERANCANGAN SISTEM ANTARMUKA BERBASIS HMI	17
3.1 Peninjauan Luas Keseluruhan Sistem (Overview).....	17
3.2 Perancangan Sistem PLC (Programmabel Logic Controller)	18
3.2.1 Sistem PLC Untuk Pengontrolan 2 mesin.....	21
3.2.2 Sistem PLC Untuk Pemberian Sinyal ke AT89S51	28
3.2.3 Sistem PLC Untuk Pengontrolan Dan Pemberian Sinyal	31
3.3 Perancangan Sistem Bridge AT89s51.....	42
3.3.1 Perancangan Sistem Bridge A.....	44
3.3.2 Perancangan sistem Bridge B.....	47
3.4 Perancangan Sistem Monitoring HMI dengan VB6	41
3.5 Perhitungan Daya	65
3.6 Sistem Pendukung Lainnya.....	66
4. PENGUJIAN DAN ANALISA SISTEM.....	49
4.1 Pengujian Sinyal Bridge.....	69
4.2 Pengujian <i>Time Process</i>	73
5. KESIMPULAN.....	77
DAFTAR REFERENSI	79
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Komponen Dasar Sistem PLC	4
Gambar 2.2 Gambar 2.2 Ladder Instruction 1	6
Gambar 2.3 Ladder Instruction 2	7
Gambar 2.4 Blok Diagram Port Serial	7
Gambar 2.5 Register TMOD	8
Gambar 2.6 Blok Diagram Mode 2	10
Gambar 2.7 Register SCON	10
Gambar 3.1 Blok Diagram Sistem Keseluruhan	17
Gambar 3.2 Komponen Sistem Antarmuka Keseluruhan	18
Gambar 3.3 Blok Diagram Sistem PLC	19
Gambar 3.4 Skema Sistem PLC	20
Gambar 3.5 Bentuk Fisik Mesin <i>Etching PCB</i>	21
Gambar 3.6 Bit Diagram Konsep PWM	22
Gambar 3.7 Pergerakan Dudukan Ferric Clorida	23
Gambar 3.8 PWM Maksimal	23
Gambar 3.9 PWM Stabil yang digunakan untuk mesin	24
Gambar 3.10 PWM Minimal	24
Gambar 3.11 Ladder Duty cycle 50%	25
Gambar 3.12 Ladder Duty cycle 66,67%	25
Gambar 3.13 Ladder Duty cycle 83,33%	26
Gambar 3.14 Ladder Program Penggabungan	28
Gambar 3.15 Gambar 3.15 Flowchart Bagian 1	31
Gambar 3.16 Gambar 3.16 Flowchart Bagian 2	32
Gambar 3.17 Bentuk Fisik Sistem PLC	33
Gambar 3.18 Blok Diagram Sistem Bridge	42

Gambar 3.19 Skema Sistem <i>Bridge</i>	43
Gambar 3.20 Flowchart Bagian 3a	44
Gambar 3.21 Pemrograman Bridge A Dengan Reads 51	45
Gambar 3.22 Flowchart Bagian 3b	47
Gambar 3.23 Pemrograman Bridge B Dengan Reads 51	48
Gambar 3.24 Blok Diagram Sistem Monitoring	51
Gambar 3.25 Monitoring HMI (<i>Run</i>)	52
Gambar 3.26 Flowchart Bagian 4	53
Gambar 3.27 Flowchart Bagian 5	54
Gambar 3.28 Flowchart Bagian 6	55
Gambar 3.29 Flowchart Bagian 7	56
Gambar 3.30 Flowchart Bagian 8	57
Gambar 3.31 Pemrograman VB6 Main 1	58
Gambar 3.32 Pemrograman VB6 Main 2	59
Gambar 3.33 Pemrograman VB6 Main 3	60
Gambar 3.34 Pemrograman VB6 Main 4	61
Gambar 3.35 Pemrograman VB6 Main 5	62
Gambar 3.36 Pemrograman VB6 Main 6	63
Gambar 3.37 Blok Diagram Sistem Output PLC Dengan Frekuensi Tinggi	68
Gambar 4.1 Diagram Waktu Serial Mode 2 (Kirim)	70
Gambar 4.2 Gambar 4.2 Diagram Waktu Serial Mode 2 (Terima)	71
Gambar 4.3 Tampilan Data Masuk Ke COM3	72
Gambar 4.4 Tampilan Data Masuk Ke COM1	72
Gambar 4.5 Tampilan <i>Time Process</i>	73
Gambar 4.6 Grafik Perbandingan TP Dan A (FIBER)	74
Gambar 4.7 Grafik Perbandingan TP Dan A (PCP)	75
Gambar 4.8 Perkiraan Jumlah Goresan PCB	76

DAFTAR TABEL

Tabel 2.1 Spesifikasi CPU PLC	5
Tabel 2.2 Catu Daya PLC	6
Tabel 2.3 Register TMOD.....	9
Tabel 2.4 Mode Operasi TMOD	9
Tabel 2.5 Pengalamatan Register SCON	11
Tabel 2.6 Datasheet Baud Rate	13
Tabel 2.7 Nilai-nilai properti even error pada CommEvent	14
Tabel 2.8 Nilai – nilai properti even komunikasi pada CommEvent.....	15
Tabel 3.1 Alamat Register Pembawa Sinyal Mesin 1 Type A.....	29
Tabel 3.2 Alamat Register Pembawa Sinyal Mesin 1 Type B.....	29
Tabel 3.3 Alamat Register Pembawa Sinyal Mesin 2 Type A.....	30
Tabel 3.4 Alamat Register Pembawa Sinyal Mesin 2 Type B.....	30
Tabel 3.5 Perhitungan Delay Bridge A	46
Tabel 3.6 Perhitungan Delay Bridge B	49
Tabel 3.7 Spesifikasi Modul Output PLC CQM1	66
Tabel 3.8 Spesifikasi Output Motor DC	67
Tabel 4.1 Data <i>Nilai Output Bridge</i>	69
Tabel 4.2 Data <i>Data Time Process</i>	74
Tabel 4.3 Perkiraan Jumlah Goresan PCB.....	76

DAFTAR SINGKATAN

AC	Alternating Current
C0	Condition 0
C1	Condition 1
C2	Condition 2
DC	Direct Current
DC (DPY)	Duty Cycle
HMI	Human Machine Interface
I/O	Input Output
M1	Machine 1
M2	Machine 2
NC	Normally Closed
NO	Normally Open
SBS	Subroutine Enter
SBN	Subrutine Define
PCB	Printed Circuit Board
PC	Personal Computer
P0.0	Port 0.0
P0.1	Port 0.1
P0.2	Port 0.2
P0.3	Port 0.3
PCP	Phenolic Cotton Papper
R.LD	Row Last Delay
R.Process	Row Process
R.Total	Row Total
R.TP	Row Time Process
SCON	Serial Control
TL	Timer Low
TH	Timer High
TMOD	Timer Mode

DAFTAR ISTILAH

<i>Baud rate</i>	Jumlah data yang ditransfer melalui sebuah interface serial
<i>Bridge</i>	Merupakan sistem AT89s51 yang bertugas membawa data dari PLC ke VB6
<i>Converter</i>	Suatu sistem atau perangkat yang berfungsi merubah sesuatu kedalam lingkungan lain, misalnya dari digital ke analog
CPU	Biasa juga dikenal dengan prosesor.
<i>Downtime</i>	Masa saat sistem mati (off)
<i>Duty Cycle</i>	Perbandingan antara waktu hidup dan perioda siklus.
Ekspansi PLC	Kemampuan CPU PLC dengan jalan menambahkan komponen hardware tertentu sehingga komputer sanggup melaksanakan suatu tugas yang tidak dapat dilaksanakan oleh CPU PLC awal.
<i>Full Duplex</i>	Kemampuan untuk berkomunikasi dan saling bertukar data dua arah secara bersamaan.
Ladder	Merupakan bentuk penulisan pemrograman PLC Syswin 3.4, yang cara kerjanya seperti relay-relay <i>logic</i>
<i>Latching</i>	Teknik penguncian yang digunakan untuk pemrograman ladder logic di input, apabila memakai tombol <i>push button</i>
<i>mnemonic</i>	Istilah yang dipakai untuk menyatakan singkatan atau memperpendek suatu bahasa atau instruksi
PLC	Digital komputer yang dapat dipergunakan untuk otomatisasi dari proses elektromekanik. Yang terdiri dari relay-relay logic
<i>Register</i>	Alat penyimpanan kecil yang terdapat dalam CPU atau prosesor dengan kecepatan akses cukup tinggi dan digunakan untuk menyimpan data dan instruksi yang sedang diproses.
<i>Time Process</i>	Waktu pada tampilan HMI yang menunjukkan lama PCB telah selesai di proses oleh mesin
<i>Visual Basic 6</i>	Merupakan software aplikasi yang menampung dan mengolah data lewat 2 Mscmm dari sistem bridge



BAB I

PENDAHULUAN

I.1 Latar Belakang Masalah

Pengembangan komputer dengan biaya rendah telah membawa kepada revolusi teknologi yaitu teknologi PLC (Programmable Logic Controller). Penggunaan PLC dimulai sejak tahun 1970 an dan menjadi pilihan utama dalam pengontrolan di industri manufaktur. Software PLC yang dijalankan pada PC dikategorikan dalam 2 hal berikut, yaitu Software PLC yang digunakan user untuk membuat program untuk menjalankan bermacam-macam kondisi input dan output (*Plant Process*), dan Software PLC yang digunakan user untuk memonitor *Plant Process* tersebut, dikenal sebagai *Human Machine Interface* (HMI) . HMI tersebut sangat membantu user dalam peningkatan kinerja mesin kedepannya, serta dapat membantu dalam *record data*.

PCB merupakan Hal yang sangat mendasar dan diperlukan dalam membuat suatu sistem rangkaian elektronika. Kalau dilakukan secara manual. Pembuatan PCB membutuhkan waktu yang lama dan tenaga yang banyak. Oleh karena itu pemakaian mesin ETCHING PCB yang dikendalikan secara otomatis memberikan manfaat yang sangat besar. Manfaat itu adalah semakin membuat mesin dapat dioperasikan dengan sedikit orang namun hasil tetap sesuai dengan harapan, menurunkan biaya produksi, mengurangi kesalahan perhitungan, ketidakakuratan dan dapat menggantikan pekerjaan-pekerjaan rutin yang dilakukan manusia.

Teknologi dasar dalam sistem pengendalian mesin ETCHING PCB adalah PLC. Penggunaan PLC ini sangat menguntungkan diantaranya adalah kemudahan dalam *troubleshooting*, sekuens control PLC dengan mudah diubah dengan pemrograman, dapat meningkatkan kualitas dan akurasi, dengan cepat mengubah dari satu produk ke produk lain. Tambahan lagi, dengan adanya kemudahan dalam *troubleshooting* maka dapat mengurangi *downtime*.

Dalam memonitor *Plant Process* yang dikerjakan oleh PLC membutuhkan *device* tambahan yang sangat kompleks. PLC juga mempunyai jenis yang sangat banyak, dan setiap perbedaan jenis PLC mempunyai sistem monitoring yang berbeda juga. Seperti

PLC jenis OMRON CQM1 harus memakai *converter* CPM1a untuk memonitoring *Plant Process*, sedangkan jenis PLC yang lain dalam satu produk memakai *converter* yang lain. Demikian pula dengan produk-produk yang dikeluarkan perusahaan lainnya seperti Mitshubishi, Festo, ABB, sangat jelas menggunakan *converter* yang berbeda.

Dengan menggunakan teknik memakai port sisa dari PLC yang *stand alone*, yang mana port sisa tersebut akan mengirim sinyal ke mikrokontroller AT89s51 dan kemudian mikrokontroller dengan kemampuannya mengirim data lewat P3.0(RXD) dan P3.1(TXD) dan kemudian data-data tersebut dapat diolah dengan VB6 maka semua jenis PLC dapat memonitor prosesnya melalui AT89s51 yang alatnya sangat mudah didapat dan murah.

I.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka penulis merumuskan permasalahan tugas akhir ini sebagai berikut:

1. Bagaimana merancang sistem PLC yang sederhana dan berguna dalam industri.
2. Bagaimana merancang sistem antarmuka antara PLC, Microcontroller dan PC sehingga HMI menjadi user-friendly.

I.3 Tujuan Penulisan

Tujuan penulisan adalah memperlihatkan suatu rancangan terpadu yang terdiri dari

1. Sistem PLC .
2. Sistem antarmuka antara HMI, PLC, Mikrokontroller, dan PC.
3. Sistem HMI yang user-friendly

I.4 Batasan Masalah

Mempertimbangkan luasnya permasalahan yang harus dipecahkan dan keterbatasan waktu pelaksanaan, maka dalam pembuatan tugas akhir ini dianggap perlu untuk membatasi permasalahan. Dalam Tugas Akhir ini, hal-hal yang akan dilakukan adalah sebagai berikut :

1. Pembuatan Program sistem PLC Omron dengan menggunakan ladder logic SYSWIN 3.4 yang digunakan untuk menjalankan 2 mesin etching PCB serta instalasi wiring cable.
2. Pembuatan Program Microcontroller AT89S51/52 sebagai Bridge / komunikasi system PLC dan Microcontroller.
3. Pengkomunikasian Microcontroller Ke Visual Basic (VB6)
4. Perancangan HMI(Human Machine Interface) dengan VB6

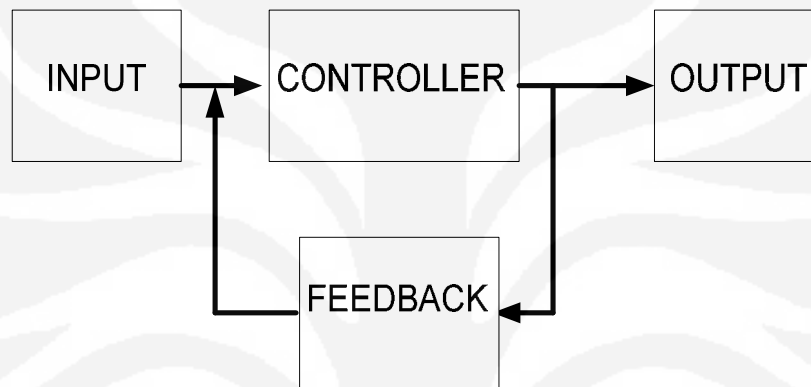
I.5 Sistematika Pembahasan

Sistematika penulisan tugas akhir ini dibagi menjadi beberapa bab. Bab satu merupakan bab pendahuluan yang berisi latar belakang, perumusan, tujuan, batasan, dan sistematika pembahasan. Bab dua berisi pembahasan teori yang menguraikan tentang pengenalan komponen-komponen utama yang terdapat pada sistem antarmuka yaitu PLC, Mikrokontroler, PC. Bab tiga berisi penggabungan sistem-sistem HMI dimulai dari interface mekanik, PLC, mikrokontroler, dan PC. Bab empat berisi pengambilan data pada output mikrokontroler (*Hyper Terminal*) yang berupa informasi awal yang nantinya akan diolah oleh VB6 sebagai tampilan HMI dan data hasil produk rangkaian elektronik pada PCB yang telah di proses oleh mesin yang berupa *Time Process*. Bab lima berisi penjelasan akhir dari tujuan penulisan disertai pengembangan sistem kedepannya, mengenai perancangan sistem antarmuka berbasis HMI untuk aplikasi mesin *Etching PCB*.

BAB II LANDASAN TEORI

2.1 PLC (Programmable Logic Controller)

Pada dasarnya setiap Sistem PLC memiliki empat komponen dasar, sebagaimana terlihat pada gambar 2.1.



Gambar 2.1. Komponen Dasar Sistem PLC

Komponen dasar dari PLC yang pertama adalah input seperti tombol tekan, limit switch, sensor seperti sensor optik, kedekatan, level, proximity. Perangkat input ini akan memberikan sinyal ke CPU dari PLC yang nantinya CPU tersebut memerintahkan output dari sistem PLC untuk bergerak. Sedangkan feedback berfungsi sebagai indikator bagi proses dari output.

Antarmuka input dan output dianalogikan sebagai indra bagi cpu PLC yaitu CPU. Sebuah PLC bekerja dengan cara menerima instruksi data dari peralatan input yang ditransfer dalam bentuk keputusan-keputusan yang bersifat logika dengan istilah bahasa *mnemonic* yang selanjutnya disimpan dalam suatu program memorinya. Selanjutnya perintah-perintah dari input akan di transfer oleh PLC ke outputnya yang selanjutnya dapat digunakan ke beban output berupa mesin-mesin atau suatu alat produksi.

2.1.1 Sourcing Dan Sinking

Istilah *sourcing* (Pensumberan) dan *sinking* (pembuangan) digunakan untuk mendeskripsikan cara penghubungan perangkat-perangkat dc ke PLC. Dengan metode *sourcing*, dengan mengasumsikan arah aliran arus yang konvensional dari positif ke negative, sebuah perangkat input menerima arus dari modul input. Artinya modul input adalah sumber (*source*) arus.

Pada metode *sinking*, dengan mengasumsikan bahwa arah aliran arus konvensional dari positif ke negative, sebuah perangkat input memberikan arus ke modul input, maksudnya modul input merupakan tempat pembuangan (*sink*) arus. Apabila arus mengalir ke modul output dari sebuah beban output dikatakan berada dalam mode *sinking*.

2.1.2 Unit CPU

Unit CPU PLC terdiri dari beberapa bagian, Berdasarkan kapasitas program, jumlah I/O points dan kecepatan pelaksanaan instruksi dapat digolongkan :

NAMA	SPESIFIKASI			Nomor Model
	I/ O point	Kapasitas Program	Kapasitas Data Program	
CJ1M- Unit CPU	160	5K	32 Kwords	CJ1M- CPU11
	160	5K	32 Kwords	CJ1M- CPU21
	320	10K	32 Kwords	CJ1M- CPU12
	640	20K	32 Kwords	CJ1M- CPU13
	320	10K	32 Kwords	CJ1M- CPU22
	640	20K	32 Kwords	CJ1M- CPU32
CJ1G- Unit CPU	1280	60K	128 Kwords	CJ1G- CPU45
	1280	30K	64 Kwords	CJ1G- CPU44
	1280	60K	128 Kwords	CJ1G- CPU45H
	1280	30K	64 Kwords	CJ1G- CPU44H
	960	20K	64 Kwords	CJ1G- CPU43H
	960	10K	64 Kwords	CJ1G- CPU42H
CJ1H- Unit CPU	2560	120K	256 Kwords	CJ1H- CPU66H
	2560	60K	128 Kwords	CJ1H- CPU65H

Tabel 2.1 Spesifikasi CPU PLC

2.1.3 Unit Catu Daya (Power Supply)

Unit catu daya berfungsi memberikan daya pada CPU dan ekspansi. Masukkan catu daya DC atau catu AC pada unit catu daya dapat dipilih sesuai kebutuhan.

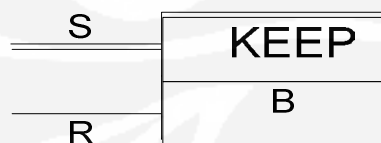
NAME	SPECIFICATIONS			
	SUPPLY VOLTAGE	OPERATING VOLTAGE RANGE	OUTPUT CAPACITY	SERVICE POWER SUPPLY
AC POWER SUPPLY UNITS	100 to 240 V AC, 50/60 Hz (Wide range)	85 to 265 V AC	5 V DC:6A 24 V DC:0.5A 30W	24 V DC:0.5A
DC POWER SUPPLY UNITS	24 V DC	20 to 28 V AC	30W 5 V DC:6A	None

Tabel 2.2 Catu Daya PLC

Pada PLC jenis Omron Cqm1 CPU 41 menggunakan AC Power Supply Unit dengan tegangan kerja pada input output AC yaitu sekitar 85 sampai 265 Vac, sedangkan tegangan input output DC yaitu sekitar 5 sampai 24 Vdc. Untuk DC Power Unit dipakai pada Tipe PLC lainnya, seperti pada tabel 2.2.

2.1.4 Ladder Syswin 3.4

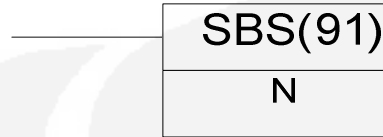
Ladder logic merupakan pemrograman dari PLC Cqm 1 yang menggunakan konsep relay- relay logic.



Gambar 2.2 Ladder Instruction 1

Instruksi diatas merupakan aplikasi dari rangkaian latching relay logic yang mana cara kerjanya adalah ketika S (set) aktif maka B (bit) on, dalam kondisi B on ketika S

diubah menjadi tidak aktif atau off nilai B tetap aktif seperti cara kerja rangkaian *latching* sederhana. Untuk membuat B menjadi kondisi 0 atau off dilakukan dengan cara mengaktifkan R atau reset.

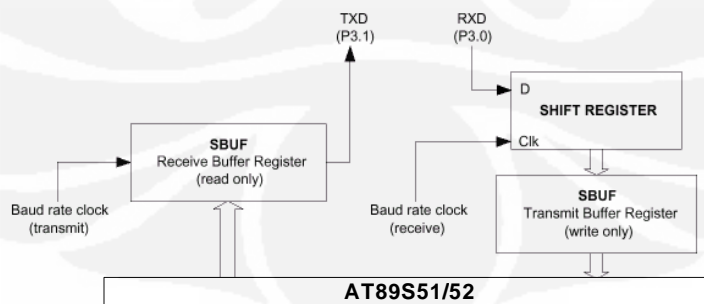


Gambar 2.3 Ladder Instruction 2

Sedangkan Instruksi ini yang kedua diatas merupakan bentuk dari aplikasi Subroutine. N merupakan Number dari subroutine yang jumlah bitnya adalah 0 sampai 255 bit. Instruksi diatas merupakan instruksi yang digunakan untuk membuat supaya jalur program menjadi lebih terstruktur, dan terutama dapat membuat lebih mudah dalam menganalisa jalannya program tersebut.

2.2 Microcontroller AT89s51

Mikrokontroller adalah suatu alat atau komponen pengontrol atau pengendali yang berukuran kecil (mikro). AT89s51/52 mempunyai *On Chip Serial Port* yang dapat digunakan untuk komunikasi data serial secara *Full Duplex* sehingga Port Serial ini masih dapat menerima data pada saat proses pengiriman data terjadi. Untuk menampung data yang diterima atau data yang akan dikirimkan, 89s51 mempunyai sebuah register yaitu SBUF yang terletak pada alamat 99H di mana register ini berfungsi sebagai buffer sehingga pada saat mikrokontroler ini membaca data yang pertama dan data kedua belum diterima secara penuh, maka data ini tidak akan hilang.

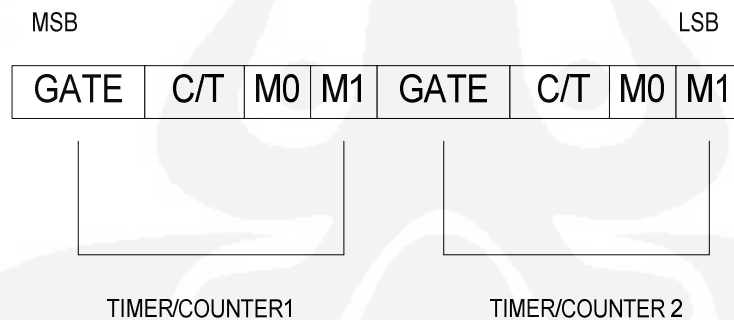


Gambar 2.4 Blok Diagram Port Serial

Pada kenyataannya register SBUF terdiri dari dua buah register yang memang menempati alamat yang sama yaitu 99H. Register tersebut adalah *Transmit Buffer Register* yang bersifat *write only* (hanya dapat ditulis) dan *Receive Buffer Register* yang bersifat *read only* (hanya dapat dibaca). Pada proses penerimaan data dari Port Serial, data yang masuk ke dalam Port Serial akan ditampung pada *Receive Buffer Register* terlebih dahulu dan diteruskan ke jalur bus internal pada saat pembacaan register SBUF. Pada proses pengiriman data ke Port Serial, data yang dituliskan dari bus internal akan ditampung pada *Transmit Buffer Register* terlebih dahulu sebelum dikirim ke Port Serial.

Port Serial 89s51 dapat digunakan untuk komunikasi data secara sinkron maupun asinkron. Komunikasi data serial secara sinkron adalah merupakan bentuk komunikasi data serial yang memerlukan sinyal clock untuk sinkronisasi di mana sinyal clock tersebut akan tersulut pada setiap bit pengiriman data sedangkan komunikasi asinkron tidak memerlukan sinyal clock sebagai sinkronisasi. Pengiriman data pada komunikasi serial 89C51 dilakukan mulai dari bit yang paling rendah (LSB) hingga bit yang paling tinggi (MSB).

2.2.1 TMOD (Timer Mode)



Gambar 2.5 Register TMOD

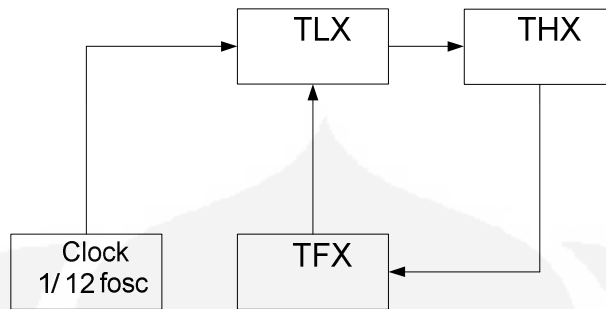
SIMBOL	KETERANGAN
GATE	Pemilih eksternal atau internal control
C/T	Pemilih Timer atau Counter
M0	Pemilih mode timer/ counter
M1	Pemilih mode timer / counter

Tabel 2.3 Register TMOD

Apabila Gate dan TRx (TR0 atau TR1 pada TCON) diberi nilai oleh user, maka timer/counter hanya beroperasi pada INTx bernilai *high*. Sedangkan nilai C/T harus diberi nilai '1' oleh program user untuk menjalankan mode counter dan diberi nilai nol untuk menjalankan mode timer. M1 dan M0 merupakan dua bit pemilih mode operasi timer/counter.

M0	M1	MODE OPERASI	
0	0	0	Timer / Counter, 13 bit
0	1	1	Timer / Counter, 16 bit
1	0	2	Timer / Counter 8, bit auto reload
1	1	3	Timer / Counter 1, berhenti

Tabel 2.4 Mode Operasi TMOD



Gambar 2.6 Blok Diagram Mode 2

Mode 2 adalah timer atau counter 8 bit dengan fasilitas auto reload. TLx bertindak sebagai timer 8 bit sedangkan THx berisi suatu nilai tertentu. Auto reload adalah fasilitas di mana nilai TLx setelah overflow tidak kembali ke 00h namun nilai TLx akan diambil dari nilai THx.

Misalkan THx berisi 47h dan TLx berisi FFh. Jika ada overflow pada TLx, nilai TLx akan dari FF ke 47h, sesuai dengan nilai THx. Proses tersebut dikerjakan secara otomatis. Mode 2 digunakan untuk menghasilkan baud rate.

2.2.2 SCON (Serial Control)



Gambar 2.7 Register SCON

BIT	ALAMAT BIT	SIMBOL	KETERANGAN
SCON7	9FH	SM0	Pemilihan Mode komunikasi serial
SCON6	9EH	SM1	Pemilihan Mode komunikasi serial
SCON5	9DH	SM2	Pemilihan Mode komunikasi multiprocessor
SCON4	9CH	SM3	Reception Enable
SCON3	9BH	SM4	Bit ke-9 yang dikirim
SCON2	9AH	SM5	Bit ke-9 yang diterima
SCON1	99H	SM6	Transmit interrupt flag
SCON0	98H	SM7	Receive interrupt flag

Tabel 2.5 Pengalamatan Register SCON

Dibawah ini merupakan penjelasan dari tabel diatas:

1. SM0 dan SM1

Merupakan pemilihan mode operasi komunikasi serial

2. SM2

Digunakan untuk mengaktifkan komunikasi multiprocessor

3. REN

Dipakai untuk mengaktifkan kemampuan port serial untuk menerima data.

4. TB8 dan RB8

Pada mode 2 dan 3, komunikasi serial bekerja dengan 9 bit data (dari 11 bit, 1 bit start, 1 bit stop), SBUF yang kapasitasnya 8 bit tidak cukup untuk keperluan ini, untuk itu disediakan tempat penampungan bit ke 9 pada saat pengiriman dan

penerimaan. Tempat tersebut yaitu TB8 yang fungsinya tempat penampungan bit ke 9 yang dikirim, dan RB8 yang fungsinya menampung bit ke 9 yang diterima port serial. Pada saat di mode 1 RB8 berfungsi untuk menampung bit-stop yang diterima. Dengan demikian apabila RB8 = 1 maka data diterima dengan benar dan sebaliknya apabila RB8 = 0 maka akan terjadi kesalahan frame (framing error).

5. TI

Berfungsi sebagai tanda bila pada port serial telah selesai mengirimkan data yang tersimpan pada SBUF. Jadi bila telah selesai mengirimkan data maka TI akan bernilai 1 dengan sendirinya. Kemudian agar bisa dipakai lagi sebagai tanda pemantau dari SBUF (saat pengiriman data) maka harus kita nolkan secara manual.

6. RI

Berfungsi sebagai tanda bila pada port serial telah selesai menerima data yang tersimpan pada SBUF. Jadi bila telah selesai mengirimkan data maka TI akan bernilai 1 dengan sendirinya. Kemudian agar bisa dipakai lagi sebagai tanda pemantau dari SBUF (saat penerimaan data) maka harus kita nolkan secara manual.

Pada mode 1 dan 3, baud rate dapat ditentukan oleh beberapa kali timer 0/1 mengalami overflow. Dengan demikian semakin sering timer mengalami overflow maka nilai baudrate makin besar. Berikut ini contoh perhitungan nilai dari TH :

1. $THx = 256 - ((\text{frek.kristal}/384)/\text{baud})$ jika bit SMOD berlogika 0.

$THx = 256 - ((\text{frek.kristal}/192)/\text{baud})$ jika bit SMOD berlogika 1.

2. Apabila baudrate = 9600, kristal = 11,0592 MHz

maka

$$THx = 256 - ((11059000/384)/9600)$$

$$= 256 - (2,93)$$

$$= 256 - 3$$

$$= 253 = 0FDh$$

Berikut ini merupakan tabel ringkasan Baud rate untuk Timer 1 sebagai Generator Baud rate:

Baud Rate	Frekuensi Kristal	SMOD	Nilai isi – ulang TH1	Baud Rate aktual	Ralat
9600	12,000 MHz	1	-7(F9H)	8923	7%
2400	12,000 MHz	0	-13 (F3H)	2404	0.16%
1200	12,000 MHz	0	-26 (F6H)	1202	0.16%
19200	11,059 MHz	1	-3(FDH)	19200	0
9600	11,059 MHz	0	-3(FDH)	9600	0
2400	11,059 MHz	0	-12(F4H)	2400	0
1200	11,059 MHz	0	-24(E8H)	1200	0

Tabel 2.6 Datasheet Baud Rate

2.3 Pengaksesan MScComm Pada Visual Basic 6

Kontrol MScComm menyediakan fasilitas komunikasi antara program aplikasi yang kita buat dengan port serial. Setiap MScComm hanya menangani satu port serial sehingga jika kita ingin menggunakan lebih dari satu port serial, kita juga harus menggunakan MScComm sebanyak port serial yang kita pakai.

2.3.1 Properti MScComm

Jumlah properti pada MScComm sangat banyak, Dibawah ini merupakan beberapa property yang penting:

1. CommPort

Digunakan untuk menentukan nomor port serial yang akan dipakai.

2. Setting

Digunakan untuk menset nilai baud rate, pariti, jumlah bit data, dan jumlah bit stop.

3. Port Open

Digunakan untuk membuka ataupun menutup port serial yang dihubungkan dengan MScComm ini.

4. Input

Digunakan untuk mengambil data string yang ada pada buffer penerima.

5. Output

Digunakan untuk menulis data string pada buffer kirim.

2.3.2 Even Pada MScComm

MScComm hanya mempunyai satu even saja, yaitu even OnComm. Even OnComm. Even OnComm dibangkitkan jika nilai dari property dari CommEvent berubah yang mengindikasikan telah terjadi even pada port serial baik even komunikasi. Maupun even error. Tabel dibawah ini adalah nilai-nilai dari properti CommEvent. Nilai property ini tidak tersedia pada saat design time, tapi hanya dibaca pada saat run time.

KONSTANTA	KETERANGAN
comEventFrame	Hardware mendeteksi adanya kesalahan framing
comEventRxParity	Hardware mendeteksi adanya kesalahan pariti
comEventRxOver	Buffer penerima mengalami over flow, tidak ada ruang kosong lagi pada buffer penerima
comEventTxFull	Buffer kirim penuh
comEventOverrun	Port mengalami overrun
comEventBreak	Sinyal Break diterima
comEventDCB	Mendapatkan kembali Device Control Block (DCB) dari port serial

Tabel 2.7 Nilai-nilai properti even error pada CommEvent

KONSTANTA	KETERANGAN
comEvsend	Jumlah karakter pada buffer kirim lebih sedikit daripada nilai properti threshold. Even ini akan dibangkitkan jika nilai pada properti Sthershold tidak diisi "0"
comEventRxParity	Telah diterima karakter sebanyak nilai properti Sthershold. Even ini akan dibangkitkan terus-menerus sampai data diambil dari buffer penerima menggunakan perintah input.
comEvCTS	Terjadi perubahan pada clear to send
comEvDSR	Terjadi perubahan pada Data Set Ready
comEvRING	Terdeteksi signal Ring
comEvEOF	Karakter End of file diterima

Tabel 2.8 Nilai – nilai properti even komunikasi pada CommEvent

Berikut ini adalah contoh penggunaan even OnComm untuk komunikasi menggunakan mikrokontroler :

```

Private static Sub MSComm1_OnComm()
Dim buffer as variant
Select Case MSComm1.Comm Event
Case comEvReceive
If MSComm1.InBufferCount >=3 then
Buffer = Cstr (MSComm1.Input)
If Mid (Buffer, 1 ,1) = "0" Then
If Mid (Buffer, 2 ,1) = "K"
Then
StatusBar1.Panels ("value").Text =
"value : " & Asc (Mid(Buffer, 3, 1))
StatusBar1.Panels ("status").Text =
"status : connect"
End if
End if
End Select
End Sub

```


Kode-kode program pada prosedur di atas akan melakukan aksi sebagai berikut:

- Mendeteksi even comEvReceive, kemudian menentukan apakah sudah diterima 3 buah karakter pada buffer penerima.
- Menentukan apakah karakter yang diterima adalah karakter “OK”. Jika karakter yang diterima adalah karakter “OK”, maka akan diubah nilai “value“ dan nilai “status” pada properti panel kontrol StatusBar1

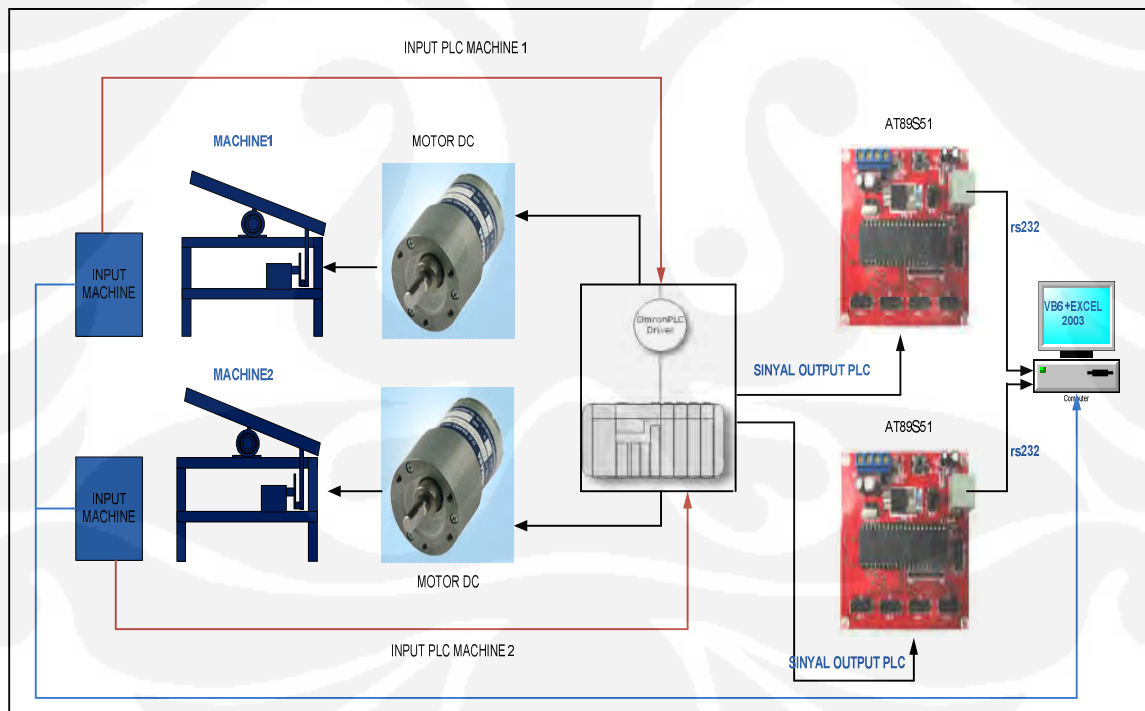
BAB III

PERANCANGAN SISTEM ANTARMUKA BERBASIS HMI

Sistem antarmuka ini dibagi menjadi 3 bagian, yaitu sistem PLC yang terhubung dengan mesin Etching PCB, dan kemudian sistem *Bridge*, yang mana sistem ini merupakan penghubung antara mesin dan PC dengan menggunakan microcontroller AT89s52, berikutnya adalah sistem Monitoring HMI pada PC atau Laptop yang sesuai standar (*compatible*).

3.1 Peninjauan Luas Keseluruhan Sistem (Overview)

Secara keseluruhan sistem Antarmuka ini adalah memanfaatkan port sisa dari PLC, yang mana port sisa ini akan dipakai untuk sinyal yang akan dikirim ke AT89s51, dan data dari AT89s51 akan dikirim ke PC, selanjutnya akan diolah oleh VB6 seperti pada blok diagram 3.1 dibawah.



Gambar 3.1 Blok Diagram Sistem Keseluruhan

Input mesin yang terdiri dari tombol-tombol dan saklar akan mengaktifkan PLC, kemudian PLC akan menggerakkan kedua Motor DC yang nantinya digunakan buat

menggerakkan mekanik mesin Etching PCB. Kondisi-kondisi dari Mesin tersebut akan ditandai dengan pengaktifan sinyal-sinyal output PLC. Selanjutnya sinyal-sinyal output PLC yang masuk ke AT89s51 akan masuk ke PC lewat port PC RS232, dan selanjutnya diolah oleh VB6 dan data dari total PCB dan failure PCB akan dikirim ke Excel 2003.

AT89s51 berjumlah dua, sehingga data dari kedua mesin dapat ditampilkan sekaligus, secara bersamaan atau dalam *time* yang sama. Gambar dibawah ini merupakan tampilan dari keseluruhan sistem, dari mesin sampai Monitoring HMI dengan menggunakan laptop Toshiba Tecra.



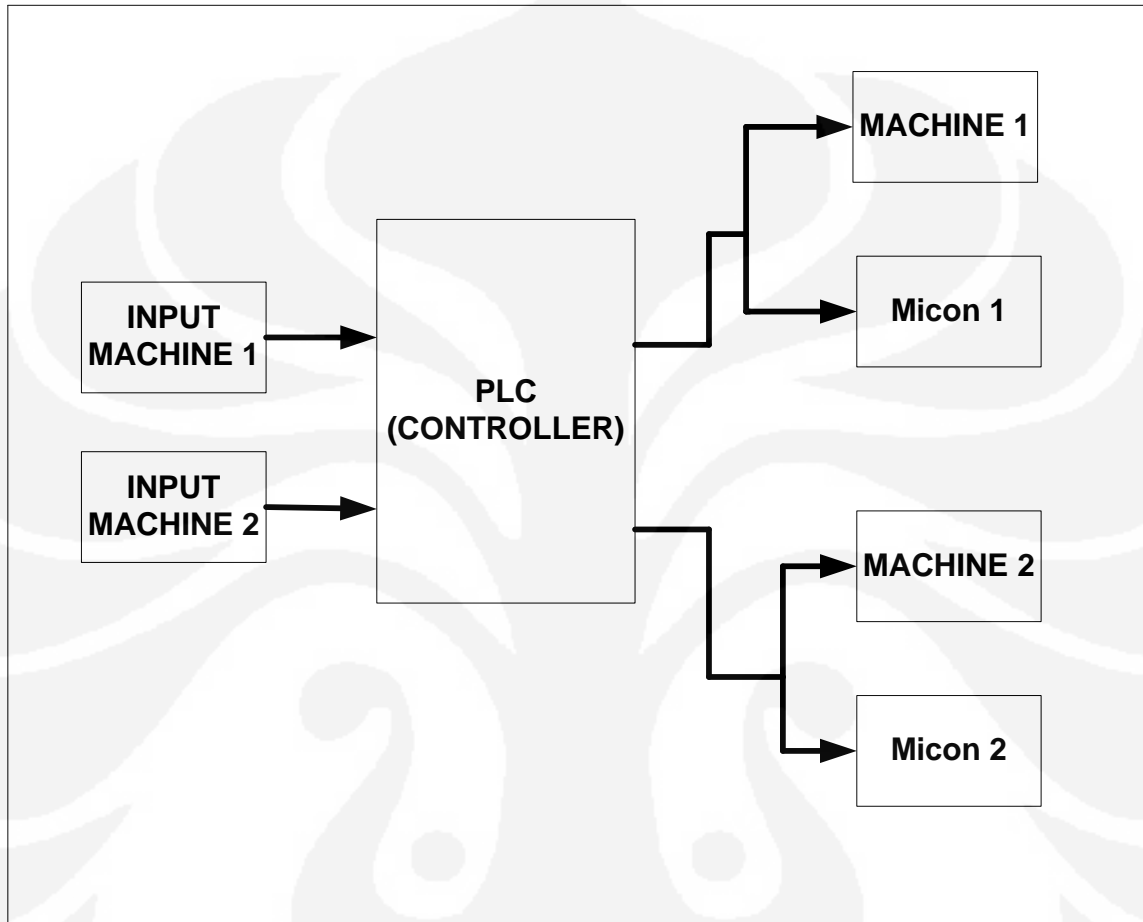
Gambar 3.2 Komponen Sistem Antarmuka Keseluruhan

3.2 Perancangan Sistem PLC (Programmable Logic Controller)

Sistem PLC ini merupakan komponen yang terpenting dalam sistem Antarmuka ini. Ada dua fungsi inti yang dikerjakan oleh PLC, yaitu :

1. Pengontrolan 2 mesin Etching PCB
2. Pemberi sinyal kepada AT89S51 yang berfungsi sebagai *bridge* data mesin ke PC.

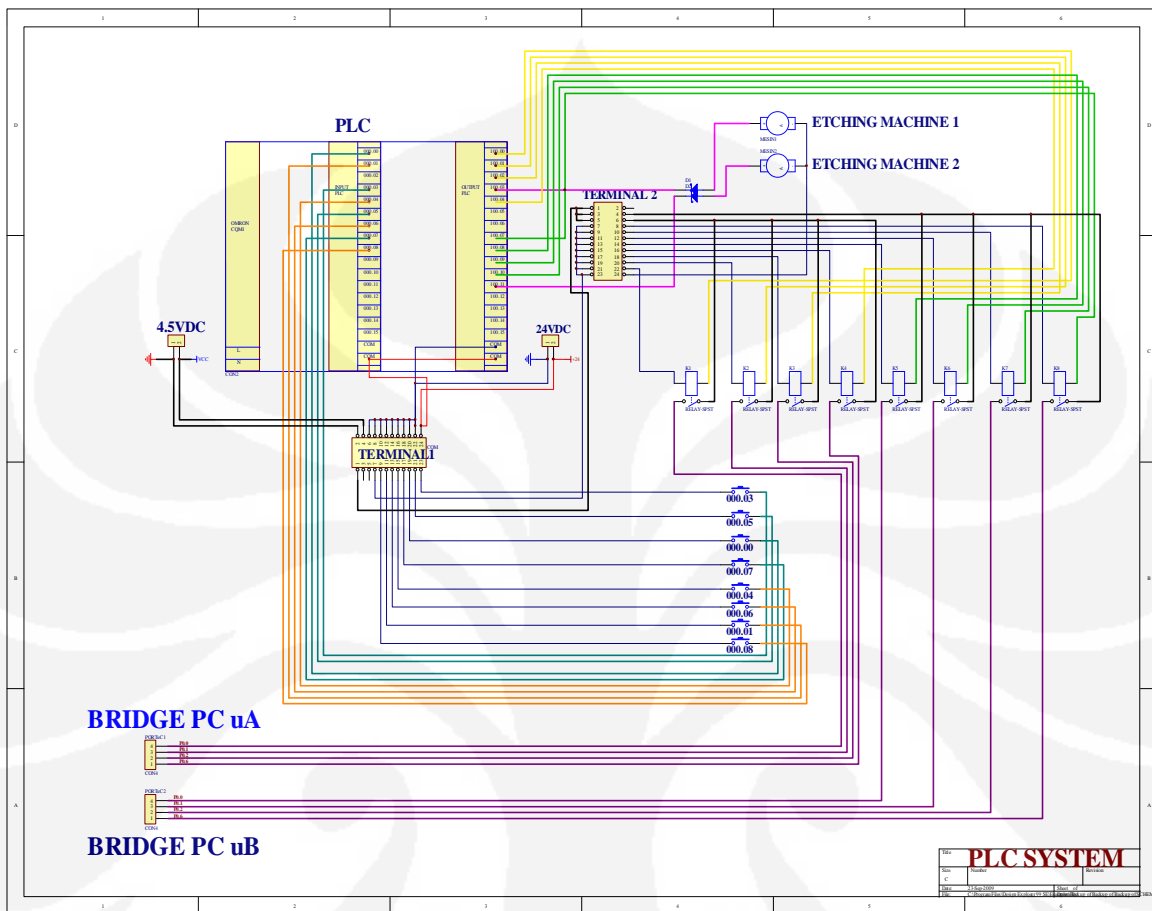
Gambar dibawah ini merupakan blok diagram dari sistem PLC yang menunjukkan sistem PLC adalah komponen dalam sistem antamuka yang mempunyai 2 fungsi.



Gambar 3.3 Blok Diagram Sistem PLC

PLC memberikan instruksi kepada mesin yang terdiri dari beberapa kondisi, yang mana kondisi-kondisi dari mesin 1 dibaca oleh Mikrokontroller 1 dan kondisi-kondisi dari mesin 2 dibaca oleh Mikrokontroller 2. Kondisi-kondisi mesin dapat terbaca oleh mikrokontroller disebabkan karena PLC terprogram untuk memberikan sinyal ke Mikrokontroller. Mikrokontroller tidak terprogram untuk mengirim signal ke PLC dapat dikatakan kondisi Mikrokontroller adalah *read* .

Gambar 3.4 merupakan schematic hardware dari sistem PLC yang sudah terangkai dengan komponen-komponennya:



Gambar 3.4 Skema Sistem PLC

Dari tampilan jalur schematic diatas dapat dijelaskan bahwa PLC memberikan 2 fungsi sekaligus. Jalur kuning dan hijau tua menjelaskan PLC terhubung dengan kelompok relay-relay 24Vdc, yang mana relay-relay tersebut akan membawa tegangan 4,5 V dc yang berguna untuk memberikan sinyal ke mikrokontroler AT89S51, dan jalur merah muda yang terkoneksi dengan mesin akan berfungsi untuk mengontrol mesin 1 dan mesin 2.

Adapun penggerak utama pada mesin ini adalah 2 motor DC yang nantinya akan di kontrol oleh PLC . Konsep dari mesin ini adalah sederhana, yaitu membuat sistem pengetsaan rangkaian PCB yang dilakukan secara otomatis sehingga operator diharapkan tidak perlu bersentuhan dengan larutan ferric clorida. Dan dengan terciptanya mesin ini diharapkan operator tidak perlu melakukan proses pengadukan yang seharusnya

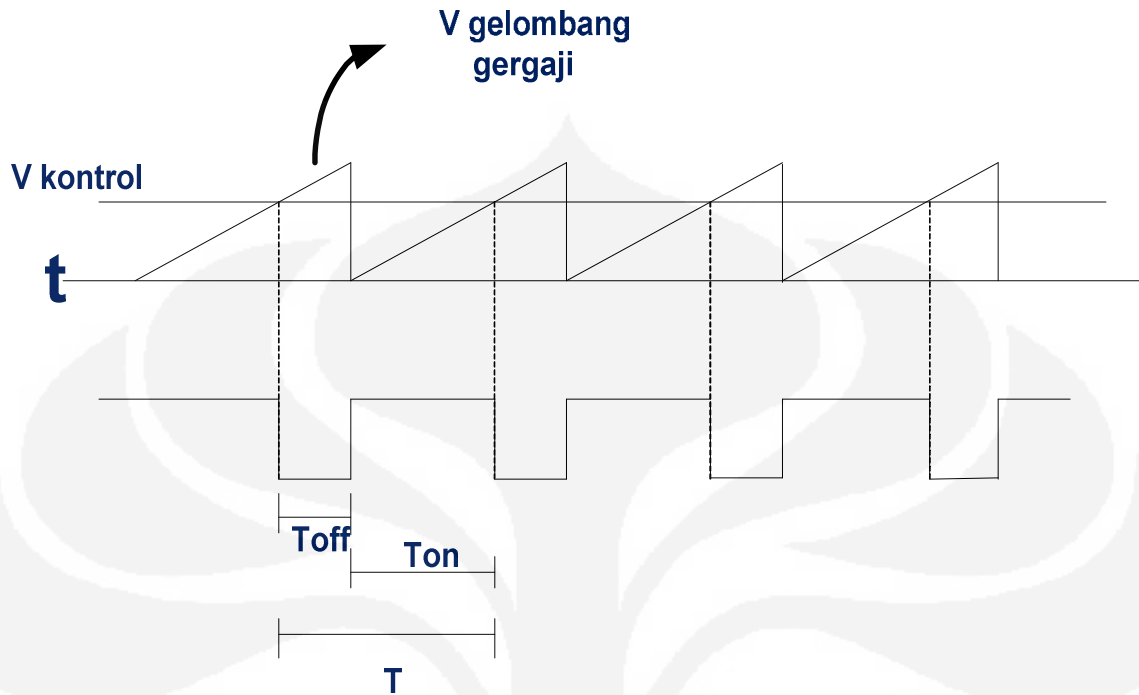
dilakukan pada proses manual. Proses pengadukan itu berfungsi supaya larutan dapat terlarut dengan air. Gambar 3.3 merupakan bentuk fisik dari 2 mesin Etching PCB.



Gambar 3.5 Bentuk Fisik Mesin *Etching PCB*

3.2.1 Sistem PLC Untuk Pengontrolan 2 Mesin

Dua mesin yang dikontrol oleh PLC menggunakan konsep PWM (Pulse Width Modulation). Konsep PWM secara dasar adalah membandingkan tegangan kontrol DC dengan tegangan gigi gergaji sehingga menghasilkan pulsa kotak dimana waktu hidup (Ton) dan waktu mati (Toff) dapat diatur melalui perbandingan kedua tegangan tersebut, dan perioda dari sinyal pulsa kotak dibuat tetap.



Gambar 3.6 Bit Diagram Konsep PWM

Sehingga didapat persamaan untuk menentukan persamaan *Duty Cycle* atau daur aktif :

$$\text{Duty Cycle} = \frac{T_{\text{on}}}{T} = T_{\text{on}} \times f$$

T_{on} = waktu hidup (s)

T = 1 siklus / perioda (s)

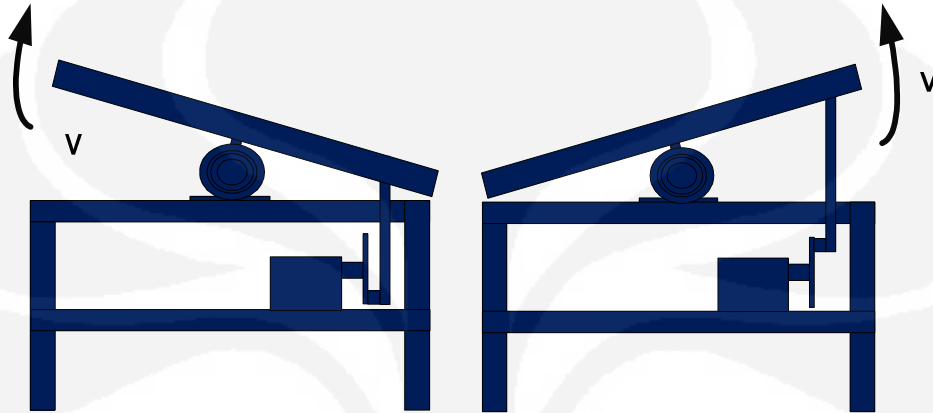
f = frekuensi (Hz)

Adapun perancangan PWM pada 2 mesin Etching PCB ini adalah melalui beberapa tahapan, yaitu:

1. Penentuan Perioda /1 siklus pada pergerakan motor
2. Pengaturan T_{on} dan T_{off} sebagai pulsa pada pergerakan motor.

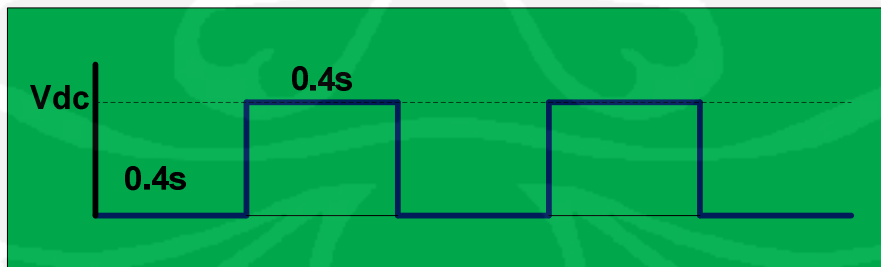
Penentuan Perioda / 1 siklus pada pergerakan motor ditentukan melalui pengujian T_{on} dan T_{off} pada perbandingan 1:1 atau perbandingan 50% : 50%. Nilai perbandingan yang dapat membuat kinerja mesin stabil merupakan nilai perioda yang dipakai pada perubahan-perubahan nilai Daur Aktif dalam satu PWM.

Pengujian perbandingan dilakukan dengan menguji nilai PWM maksimal dan nilai PWM minimal . Nilai dari PWM maksimal adalah 0,4s : 0,4s, sedangkan nilai PWM minimal didapat 0,2s : 0,2s. Nilai PWM maksimal dan minimal ini didapat dari hasil percobaan dan analisa. Gambar dibawah merupakan pergerakan dudukan larutan untuk PCB.

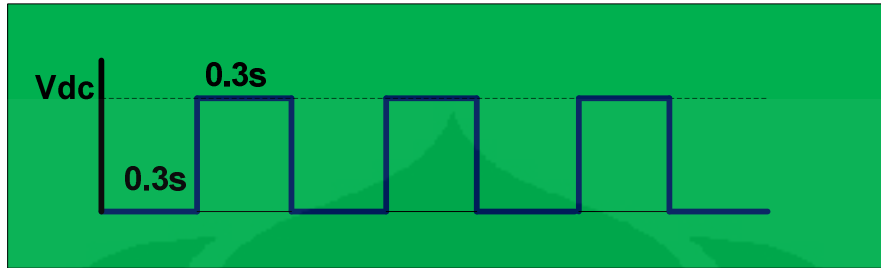


Gambar 3.7 Pergerakan Dudukan Ferric Clorida

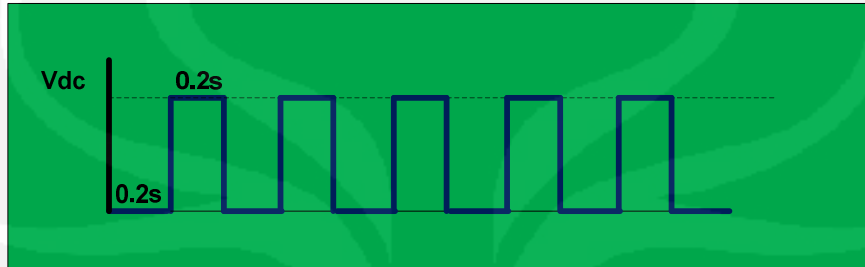
Untuk Nilai perbandingan Ton dan Toff pada 0,5s : 0,5s mempunyai nilai V yang terlalu besar ($V \gg$) sehingga proses produk PCB yang akan dietching tidak normal. Sedangkan untuk nilai perbandingan Ton dan Toff pada 0,1s : 0,1s mempunyai nilai V yang terlalu kecil ($V \ll$) sehingga motor tidak dapat menggerakkan dudukan larutan ferric clorida. Oleh karena itu didapat nilai PWM maksimal adalah 0,4s : 0,4s sedangkan nilai PWM minimal adalah 0,2s : 0,2s dan nilai PWM yang normal dan stabil adalah 0,3s berbanding 0,3s. Sehingga didapat nilai Perioda dalam PWM yang dipakai adalah Ton dan Toff yang stabil yaitu 0,6s ($Ton + Toff$).



Gambar 3.8 PWM Maksimal



Gambar 3.9 PWM Stabil yang digunakan untuk mesin



Gambar 3.10 PWM Minimal

Perubahan nilai-nilai pada perbandingan T_{on} dan T_{off} atau 0,3s : 0,3s atau dapat dikatakan presentasi daur aktif 50% terjadi dua perubahan, yaitu :

1. Menjadi 0,4s : 0,2s sehingga presentasi daur aktifnya adalah:

$$\% \text{Daur aktif} = \frac{T_{on}}{T} \times 100\%$$

$$\% \text{Daur aktif} = \frac{0,4}{0,6} \times 100\% \longrightarrow 66,67\%$$

2. Menjadi 0,5s : 0,1s sehingga presentasi daur aktifnya adalah:

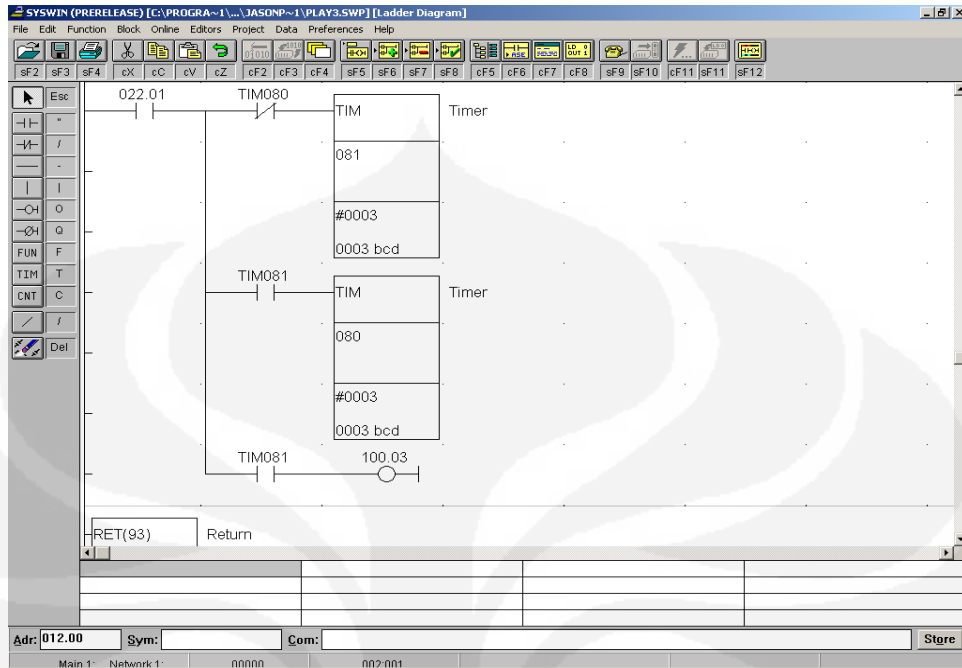
$$\% \text{Daur aktif} = \frac{T_{on}}{T} \times 100\%$$

$$\% \text{Daur aktif} = \frac{0,5}{0,6} \times 100\% \longrightarrow 83,33\%$$

Adapun potongan program pada SYSWIN 3.4 yang menyatakan beberapa perubahan daur aktif pada sistem PLC adalah :

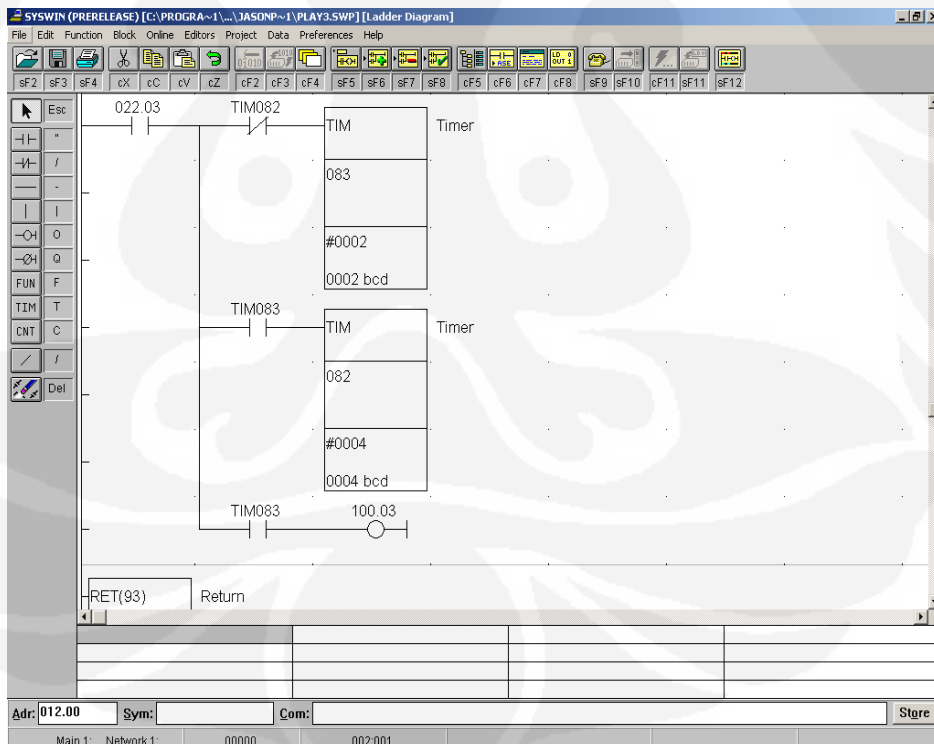
1. Daur aktif 50% dengan nilai perbandingan T_{on} dan T_{off} adalah 0,3s : 0,3s.

Gambar dibawah ini merupakan contoh potongan program ladder :



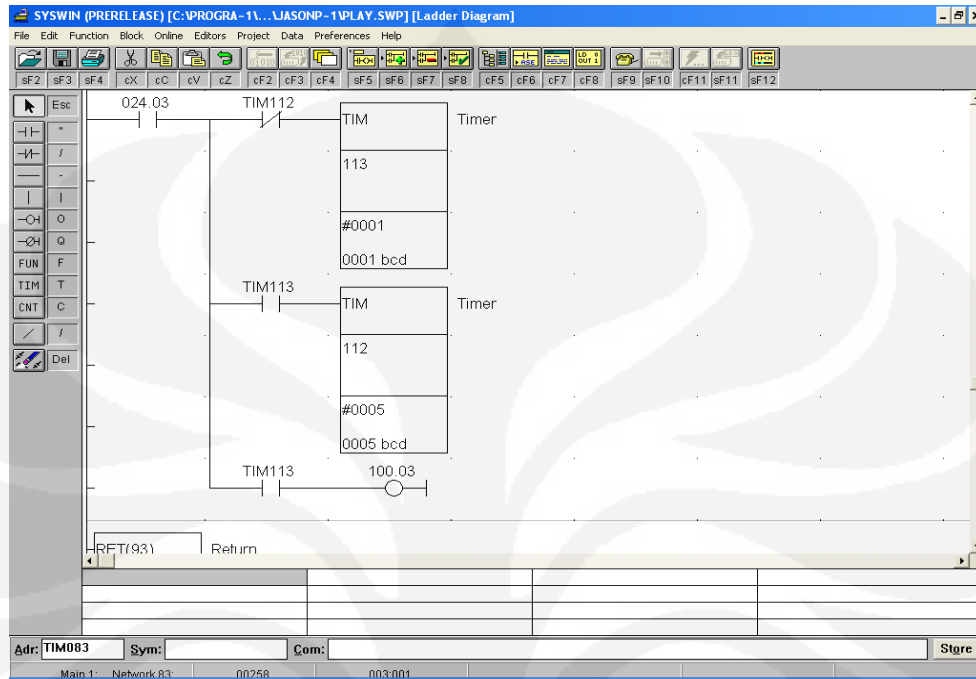
Gambar 3.11 Ladder Duty cycle 50%

2. Daur Aktif 66,67% dengan nilai perbandingan Ton dan Toff adalah 0,4s : 0,2s. Gambar 3.10 adalah program laddernya.



Gambar 3.12 Ladder Duty cycle 66,67%

3. Daur Aktif 83,33% dengan nilai perbandingan Ton dan Toff adalah 0,5s : 0,1s. Gambar 3.11 adalah program laddernya.



Gambar 3.13 Ladder Duty cycle 83,33%

Cara kerja dari ladder daur aktif 50% adalah ketika input 022.01 kondisi NC (Normally Closed) maka timer 081 akan aktif selama 0,3s (Toff). Setelah nilai dari 0,3s dari timer 081 menjadi 0 maka input ladder TIM081 akan menjadi NC sedangkan input ladder TIM080 akan menjadi NO, sehingga langsung mengaktifkan timer 080 selama 0,3s (Ton) dan output mesin 1 atau 100.03 akan aktif. Setelah nilai dari 0,3s dari timer 080 menjadi 0 maka input ladder TIM081 kembali menjadi NO dan input ladder TIM080 akan menjadi NC. Selanjutnya proses akan berulang kembali (*looping*) sampai bagian subroutine program ini selesai, maka akan pindah ke subroutine berikutnya. Daur aktif ini disebut kondisi 0 (C0) dalam proses pengiriman sinyal tahap selanjutnya. Kondisi C0 pada proses etching PCB merupakan tahap awal dalam pergerakan mesin yang fungsinya sebagai pengaduk otomatis serbuk Ferric Clorida dengan air panas sekitar 120ml sampai 140ml. Supaya semua serbuk dapat terlarut dalam air. Daur aktif 50% ini juga dipakai pada saat tahapan proses yang kelima atau terakhir yang digunakan sebagai delay 10s untuk memberhentikan mesin, hal ini dilakukan untuk mencegah apabila mesin

mengerjakan bagian tertentu dari layout yang susah diproses sehingga membutuhkan waktu lama. Maka dibuatlah suatu sistem delay 10s yang tetap pada daur aktif 50%.

Cara kerja dari ladder daur aktif 66,67% adalah ketika input 022.03 kondisi NC (Normally Closed) maka timer 083 akan aktif selama 0,2s (Toff). Setelah nilai dari 0,2s dari timer 083 menjadi 0 maka input ladder TIM083 akan menjadi NC sedangkan input ladder TIM082 akan menjadi NO, sehingga langsung mengaktifkan timer 082 selama 0,4s (Ton) dan output mesin 1 atau 100.03 akan aktif. Setelah nilai dari 0,4s dari timer 080 menjadi 0 maka input ladder TIM083 kembali menjadi NO dan input ladder TIM082 akan menjadi NC. Selanjutnya proses akan berulang kembali (*looping*) sampai bagian subroutine program ini selesai, maka akan pindah ke subroutine berikutnya. Daur aktif ini disebut kondisi 1 (C1) dalam proses pengiriman sinyal tahap berikutnya. Seperti pada kondisi C1 dengan daur aktif 66,67% atau dengan peningkatan yang tidak terlalu jauh, kondisi C1 semakin membuat serbuk terlarut kedalam air. Selain itu kondisi C1 adalah kondisi yang sangat berguna untuk mengurangi gaya gesekan pada *bearing* atau navigasi dudukan larutan Ferric Clorida, sehingga dapat memperkecil kerusakan pada bagian tersebut. Gaya gesekan yang maksimal berlangsung 360s pada PCB tipe A maka dapat dilakukan hanya 144s, selebihnya dikerjakan oleh duty cycle 50% dan 66,67%. Selain itu hal terpenting adalah perubahan duty cycle ini dilakukan untuk mengurangi konsumsi daya. Sehingga dengan adanya konsumsi daya yang kecil maka penambahan mesin bisa dilakukan untuk kedepannya.

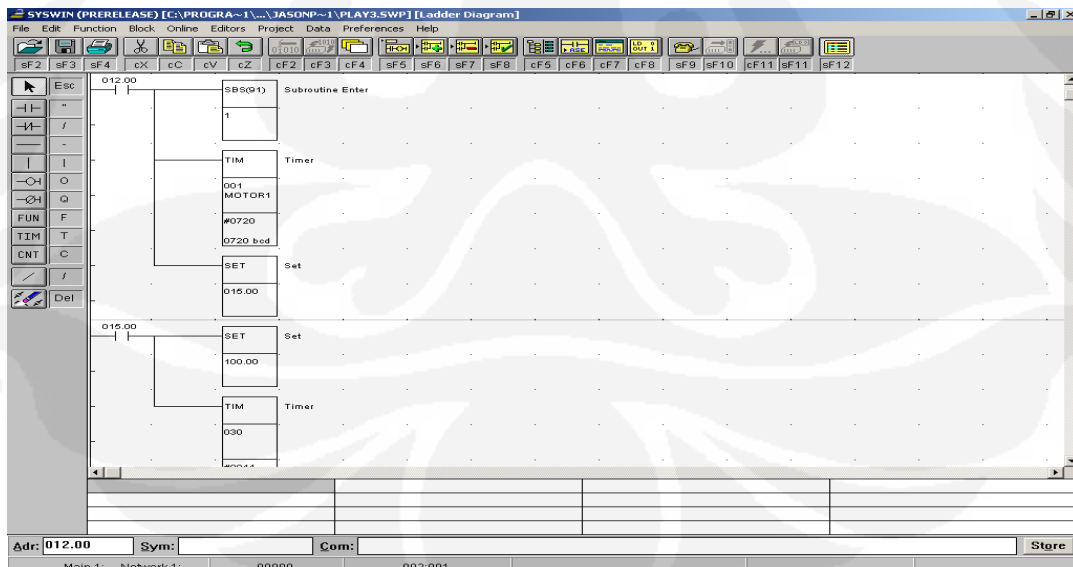
Cara kerja dari ladder daur aktif 83,33% adalah ketika input 024.03 kondisi NC (Normally Closed) maka timer 113 akan aktif selama 0,1s (Toff). Setelah nilai dari 0,1s dari timer 113 menjadi 0 maka input ladder TIM113 akan menjadi NC sedangkan input ladder TIM112 akan menjadi NO, sehingga langsung mengaktifkan timer 112 selama 0,5s (Ton) dan output mesin 1 atau 100.03 akan aktif. Setelah nilai dari 0,5s dari timetr 080 menjadi 0 maka input ladder TIM113 kembali menjadi NO dan input ladder TIM112 akan menjadi NC. Selanjutnya proses akan berulang kembali (*looping*) sampai bagian subroutine program ini selesai, maka akan pindah ke subroutine berikutnya. Daur aktif ini disebut kondisi 2 (C2) dalam proses pengiriman sinyal tahap berikutnya. Pada kondisi inilah inti dari proses etching yang berlangsung sekitar 144 detik. Pada tahapan ini sudah

bisa diperkirakan bahwa serbuk sudah terlarut didalam air sehingga proses ini lebih mengutamakan penggoresan larutan terhadap layout rangkaian elektronik yang sudah tergambar pada PCB.

Dan untuk cara kerja sistem PLC yang merupakan bagian pertama dalam sistem interfacing berbasis HMI ini dapat dijelaskan pada flowchart 3.13. Yang mana dari flowchart digambarkan bahwa PLC mempunyai peranan dalam menjalankan 2 mesin sekaligus, atau hanya menjalankan 1 mesin saja.

3.2.2 Sistem PLC Untuk Pemberian Sinyal Ke AT89S51

Konsep Pemberian sinyal dari PLC ke Mikrokontroler adalah dengan cara menggabungkan ladder program perubahan kondisi mesin dengan ladder program pengiriman sinyal output PLC melalui port output PLC, dalam sistem ini port yang dipakai adalah port 100.00, 100.01, 100.02 pada mesin 1 dan 100.08, 100.09, 100.10 pada mesin 2. Sehingga pada saat mesin 1 masuk ke kondisi 0 maka PLC akan mengaktifkan port 100.00, pada saat kondisi 1 maka PLC akan mengaktifkan port 100.01, demikian seterusnya. Dibawah ini merupakan penggabungan program ladder perubahan kondisi mesin dan ladder pengiriman sinyal output.



Gambar 3.14 Ladder Program Penggabungan Sinyal

Ketika input ladder 012.00 menjadi NC maka mesin 1 akan hidup dalam kondisi 0 (C.0) melalui panggilan Subroutine (SBS) 1 dan sekaligus mengeset 015.00 yang fungsinya untuk menghidupkan port 100.00.

Adapun jumlah sinyal dari setiap proses pengiriman sinyal dari satu mesin adalah 4, yaitu sinyal kondisi 0,1,2,dan 6 (failure). Dan jumlah tahapan dalam setiap proses mesin adalah 6 yaitu kondisi 0,1,2,1,0, dan 6. Tabel dibawah ini menunjukkan alamat register pada program SYSWIN 3.4.

NO	SIGNAL TO uC a	SET ADRESS REGISTER	SUBROUTINE	PORT OUTPUT PLC
1	P0.0 - C0	015.00	1	100.00
2	P0.1 – C1	015.01	3	100.01
3	P0.2 – C2	015.02	5	100.02
4	P0.1 – C1	015.03	7	100.01
5	P0.0 – C0	015.04	9	100.00
6	P0.6			100.04

Tabel 3.1 Alamat Register Pembawa Sinyal
Mesin 1 Type A

NO	SIGNAL TO uC a	SET ADRESS REGISTER	SUBROUTINE	PORT OUTPUT PLC
1	P0.0 - C0	015.05	11	100.00
2	P0.1 – C1	015.06	13	100.01
3	P0.2 – C2	015.07	15	100.02
4	P0.1 – C1	015.08	17	100.01
5	P0.0 – C0	015.09	19	100.00
6	P0.6			100.04

Tabel 3.2 Alamat Register Pembawa Sinyal
Mesin 1 Type B

NO	SIGNAL TO uC b	SET ADDRESS REGISTER	SUBROUTINE	PORT OUTPUT PLC
1	P0.0 - C0	016.00	2	100.08
2	P0.1 – C1	016.01	4	100.09
3	P0.2 – C2	016.02	6	100.10
4	P0.1 – C1	016.03	8	100.09
5	P0.0 – C0	016.04	10	100.08
6	P0.6			100.05

Tabel 3.3 Alamat Register Pembawa Sinyal
Mesin 2 Type A

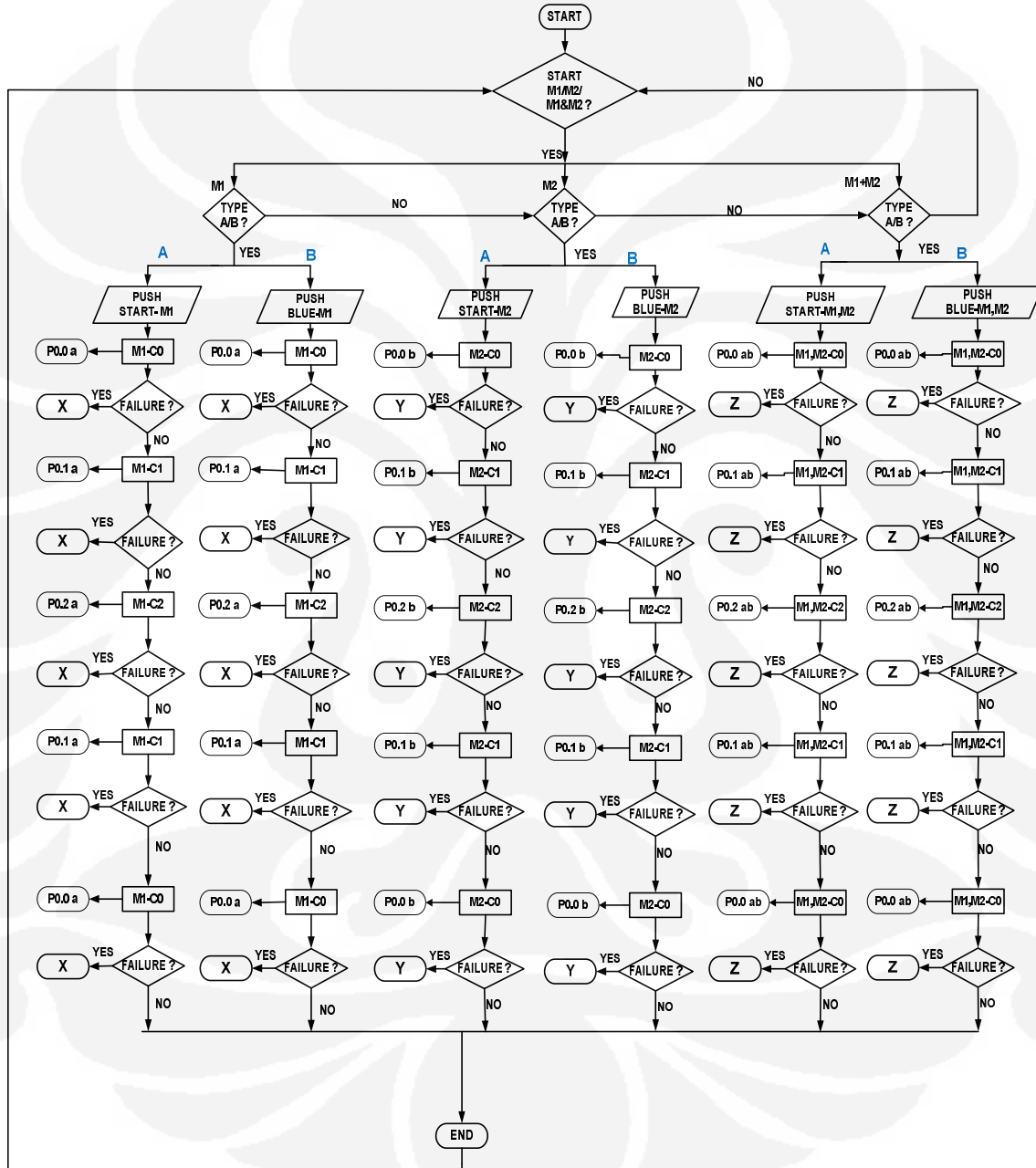
NO	SIGNAL TO uC b	SET ADDRESS REGISTER	SUBROUTINE	PORT OUTPUT PLC
1	P0.0 - C0	016.05	12	100.08
2	P0.1 – C1	016.06	14	100.09
3	P0.2 – C2	016.07,	16	100.10
4	P0.1 – C1	016.08	18	100.09
5	P0.0 – C0	016.09	20	100.08
6	P0.6			100.06

Tabel 3.4 Alamat Register Pembawa Sinyal
Mesin 2 Type B

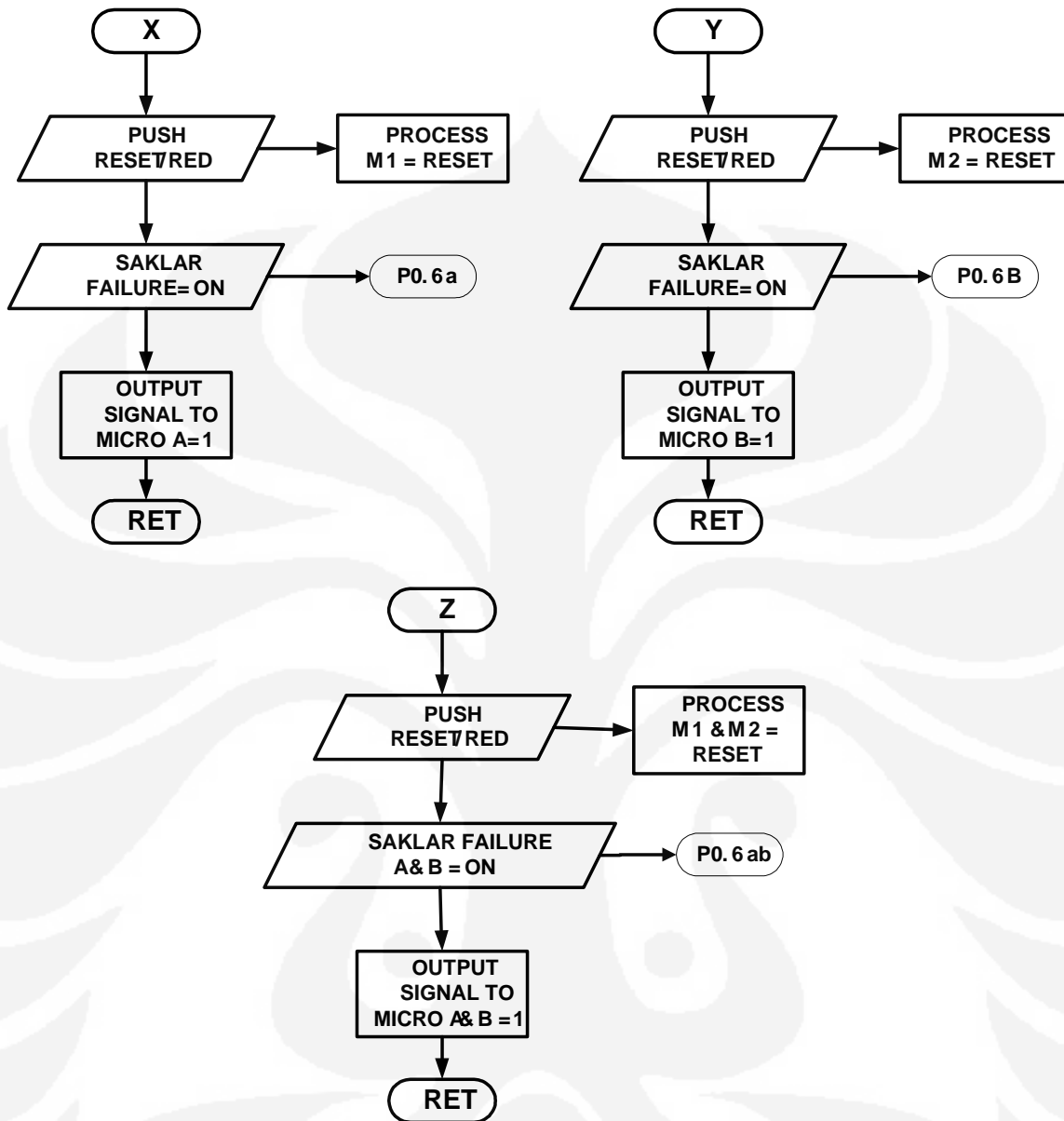
Dari tabel-tabel diatas dapat diketahui bahwa setiap proses 1 mesin dan 1 type membutuhkan 6 tahapan yaitu C0,C1,C2,C1,C0, dan tahapan berikutnya adalah hasil failure apabila ada kesalahan (C6) namun apabila tidak ada kesalahan proses berlangsung 5 tahapan.

3.1.3 Pemrograman PLC Sebagai Sistem Pengontrolan Dan Pemberi Sinyal

Untuk sistem PLC secara keseluruhan merupakan penggabungan dari sistem pengontrolan 2 mesin dan pemberian sinyal ke AT89S51 yang dirangkai menjadi 1 *source code* ladder logic. Flowchart 3.15 merupakan konsep keseluruhan dari sistem PLC.



Gambar 3.15 Flowchart Bagian 1

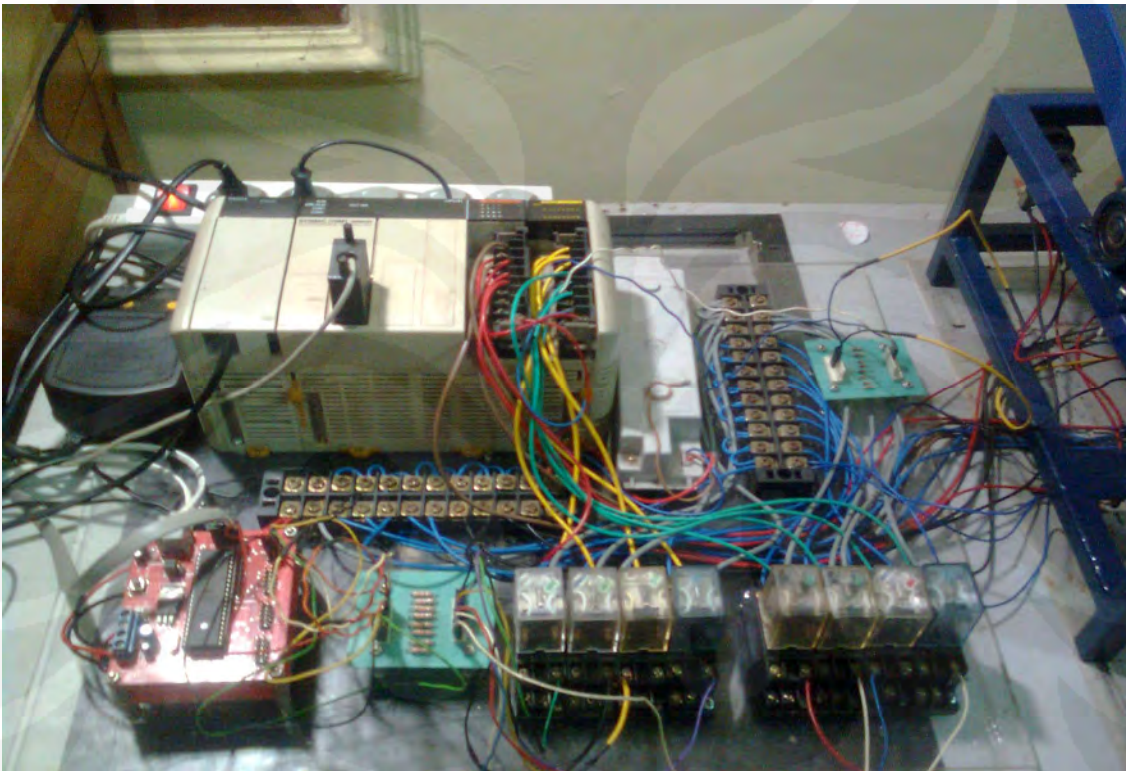


Gambar 3.16 Flowchart Bagian 2

Kemampuan Sistem PLC dapat menjalankan 2 mesin sekaligus atau hanya 1 mesin yang dijalankan. Ketika hanya mesin 1 yang dijalankan maka PLC memberikan pilihan type PCB yang mana Type A adalah PCB jenis fiber, sedangkan PCB Type B adalah PCB jenis Phenolic Cotton Paper. Setelah itu dengan menekan tombol start maka mesin mulai memproses melalui C0,C1,C2,C1,dan C0 lagi. Setiap melawati Tahapan ada 2 yang dikerjakan PLC yaitu mengirim data-data tahapan proses tadi (C0,C1,C2) yang

nantinya data-data ini diolah oleh PC, dan kemudian setiap tahapan PLC selalu menunggu instruksi berhenti yang akan dikerjakan oleh tombol reset apabila ada kesalahan dalam proses. Dan ketika tombol reset berhenti user dapat mengaktifkan tombol failure pada mesin dan sinyal failure itu akan dibawa oleh Micro untuk diproses di PC.

Gambar dibawah ini merupakan bentuk fisik dari sistem PLC Omron yang telah terhubung dengan sistem lainnya.



Gambar 3.17 Bentuk Fisik Sistem PLC

Konsep Dalam sistem pemrograman ladder ini adalah penggabungan dari ladder pengontrol mesin 1 untuk tipe A, ladder pengontrol mesin 1 untuk tipe B, ladder pengontrol mesin 2 untuk tipe A, dan ladder pengontrol mesin 2 untuk tipe B yang disatukan sehingga menjadi hanya 1 ladder yang terstruktur. Ladder pada lampiran A sampai M adalah bentuk dari ladder pemrograman sistem ini.

Proses pergerakan dari mesin 1 tipe A (fiber PCB) dijalankan oleh ladder bagian 1 dan ladder bagian 2 pada lampiran A dan B, yang mana pada bagian tersebut pengontrolan mesin 1 disertai dengan pemberian sinyal ke mikrokontroler AT89S51 dilakukan. Untuk pengontrolan mesin 1 ditandai dengan pemanggilan subroutine 1,3,5,7,9, yang dilakukan dengan instruksi ladder logic SBS(91). Adapun fungsi dari SBS(91) adalah memanggil subroutine yang ditandai dengan instruksi SBN(92), yang terletak pada ladder Lampiran H dan Lampiran I. Pada bagian inilah PWM dengan daur aktif 50%, 66,67%, 83,33% di mesin 1 tipe A dijalankan. Ketika Tombol start M1 (000.03) di aktifkan maka register 12.00 aktif terkunci (latching) yang artinya adalah proses yang akan dijalankan pada register 12.00 tidak akan berhenti atau berulang, walaupun tombol 000.03 tertekan. Namun register 12.00 akan berhenti apabila tombol input 000.06 ditekan. Setelah register 12.00 aktif maka akan ada 3 proses yang dilakukan yaitu SBS 1 aktif, timer 001 aktif, dan register 15.00 aktif. Instruksi SBS 1 adalah pengaktifan instruksi SBN 1 yang mana didalam instruksi ini PWM mesin 1 pada kondisi daur aktif 50% di proses (C0). Mesin 1 pada daur aktif 50 % ini diproses selama timer 001 aktif, yaitu sekitar 72 detik. Pada ladder 3.1.3 pada bagian 1 dituliskan pada intruksi timer 001 dengan cara #0720, yang berarti nilai timer pada omron 720 dibagi konstanta 10 yaitu 72 detik. Sedangkan pengaktifan register 015.00 mempunyai fungsi yang sudah dijelaskan pada tabel 3.1.2 yaitu pembawa sinyal ke mikrokontroler dengan cara mengaktifkan port 100.00 dan kemudian port 100.00 ini akan menhidupkan relay K1 aktif selama sekitar 1,1 detik (#0011). Setelah 1,1 detik maka output kontak ladder NO TIM30 akan aktif yang membuat output port 100.00 mati (reset). Dan ketika tim 001 selesai melaksanakan instruksinya maka input ladder NO TIM001 akan aktif sehingga proses dari ladder memasuki tahap kedua yaitu pengaktifan instruksi SBS 3. Dengan aktifnya SBS 3, maka instruksi SBN 3 akan memproses mesin 1 menjadi kondisi daur aktif 66,67% (C1). Mesin 1 pada daur aktif 66,67% ini diproses selama timer 003 aktif, yaitu sekitar 72 detik. Pengaktifan sinyal pada proses ini dilakukan oleh register 15.01, sehingga port 100.01 aktif, selama 1,1 detik. Setelah timer 003 selesai melaksanakan instruksinya maka output kontak ladder NO TIM003 akan aktif sehingga proses dari ladder memasuki tahap ketiga yaitu pengaktifan instruksi SBS 5. Sehingga instruksi SBN

5 aktif, lalu memproses mesin 1 menjadi kondisi daur aktif 83.33% (C2). Mesin 1 pada daur aktif 83,33% ini diproses selama timer 005 aktif, yaitu sekitar 134 detik (#1340). Pengaktifan sinyal pada proses ini dilakukan oleh register 15.02, sehingga port 100.02 aktif selama 1,1detik. Setelah timer 005 selesai melaksanakan instruksinya maka output kontak ladder NO TIM005 akan aktif sehingga proses dari ladder memasuki tahap keempat yaitu pengaktifan instruksi SBS 7, dan secara otomatis instruksi SBN 7 aktif, lalu memproses mesin 1 menjadi kondisi daur aktif 66,67% (C1). Mesin 1 pada daur aktif 66,67% diproses selama selama timer 007 aktif, yaitu sekitar 72 detik. Pengaktifan sinyal dilakukan oleh register 15.01, sehingga port 100.01 kembali aktif selama 1,1 detik. Setelah timer 007 selesai melaksanakan instruksinya maka output kontak ladder NO TIM007 akan aktif sehingga proses dari ladder memasuki tahap terakhir dari mesin 1 tipe A ini atau tahap kelima yaitu pengaktifan instruksi SBS 9, dan secara otomatis instruksi SBN 9 aktif, lalu memproses mesin 1 menjadi kondisi daur aktif 50%(C0). Mesin 1 pada tahap kelima ini diproses oleh timer 009 selama 10 detik. Berbeda dari tahap-tahap sebelumnya pada tahapan ini mesin berada pada tahapan delay pemberhentian dari proses. Setelah timer 009 selesai melaksanakan instruksinya maka output kontak ladder NO TIM009 akan aktif sehingga proses dari *etching PCB* oleh mesin 1 selesai.

Perhitungan waktu dari mulai tahapan pertama, kedua, ketiga, keempat, dan kelima berasal dari pembagian dari perkiraan waktu total selesainya proses *etching PCB*. Yang diperkirakan waktu total untuk tipe A adalah 360s. Nilai dari 360 detik adalah nilai yang didapatkan dari beberapa kali percobaan pada rangkaian yang akan diproses oleh mesin. Sehingga tahap pertama berlangsung selama 72 detik, tahap kedua berlangsung selama 72 detik, tahap ketiga adalah inti dari proses maka waktu ditambah dua kali menjadi 114 detik, tahap keempat kembali menjadi 72 detik, dimana pada tahap ini diperkirakan rangkaian PCB sudah selesai diproses dan pada tahap kelima adalah tahap pemberhentian mesin yang diberi delay 10 detik.

Pemrograman sinyal failure terlihat pada ladder baris ketiga yang proses kerjanya adalah ketika input kontak 000.00 menjadi *Normally Closed*, maka output port pada PLC yaitu 100.04 akan aktif dan menghidupkan relay K4 untuk membawa sinyal ke

mikrokontroler AT89S51. Proses perubahan input kontak normaly closed menjadi normally open dilakukan manual oleh saklar input pada mesin.

Proses pergerakan dari mesin 1 tipe B (Phenolic Cotton Paper) dijalankan oleh ladder bagian 3 dan bagian 4 pada lampiran C dan D, yang mana pada bagian tersebut pengontrolan mesin 1 disertai dengan pemberian sinyal ke mikrokontroler AT89S51 dilakukan. Untuk pengontrolan mesin 1 ditandai dengan pemanggilan subroutine 11,13,15,17,19, yang juga dilakukan dengan instruksi ladder logic SBS(91). Adapun fungsi dari SBS(91) adalah memanggil subroutine yang ditandai dengan instruksi SBN(92), yang terletak pada ladder pada lampiran I dan J. Pada bagian inilah PWM dengan daur aktif 50%, 66,67%, 83,33% di mesin 1 tipe B dijalankan. Ketika Tombol start M1 (000.05) di aktifkan maka register 13.00 aktif terkunci (latching) yang artinya adalah proses yang akan dijalankan pada register 13.00 tidak akan berhenti atau berulang, walaupun tombol 000.05 tertekan. Namun register 13.00 akan berhenti apabila tombol input 000.06 ditekan. Setelah register 13.00 aktif maka akan ada 3 proses yang dilakukan yaitu SBS 11 aktif, timer 011 aktif, dan register 15.05 aktif. Instruksi SBS 11 adalah pengaktifan instruksi SBN 11 yang mana didalam instruksi ini PWM mesin 1 pada kondisi daur aktif 50% di proses (C0). Mesin 1 pada daur aktif 50 % ini diproses selama timer 011 aktif, yaitu sekitar 60 detik. Pada ladder 3.1.3 pada bagian 3 dituliskan pada intruksi timer 011 dengan cara #0600, yang berarti nilai timer pada omron 600 dibagi konstanta 10 yaitu 60 detik. Sedangkan pengaktifan register 015.05 mempunyai fungsi yang sudah dijelaskan pada tabel 3.1.2 yaitu pembawa sinyal ke mikrokontroler dengan cara mengaktifkan port 100.00 dan kemudian port 100.00 ini akan menhidupkan relay K1 aktif selama sekitar 1,1 detik (#0011). Setelah 1,1 detik maka output kontak ladder NO TIM35 akan aktif yang membuat output port 100.00 mati (reset). Dan ketika tim 011 selesai melaksanakan instruksinya maka input ladder NO TIM001 akan aktif sehingga proses dari ladder memasuki tahap kedua yaitu pengaktifan instruksi SBS 13. Dengan aktifnya SBS 13, maka instruksi SBN 13 akan memproses mesin 1 menjadi kondisi daur aktif 66,67% (C1). Mesin 1 pada daur aktif 66,67% ini diproses selama timer 013 aktif, yaitu sekitar 60 detik. Pengaktifan sinyal pada proses ini dilakukan oleh register 15.06, sehingga port 100.01 aktif, selama 1,1 detik. Setelah timer 013 selesai melaksanakan

instruksinya maka output kontak ladder NO TIM013 akan aktif sehingga proses dari ladder memasuki tahap ketiga yaitu pengaktifan instruksi SBS 15. Sehingga instruksi SBN 15 aktif, lalu memproses mesin 1 menjadi kondisi daur aktif 83,33% (C2). Mesin 1 pada daur aktif 83,33% ini diproses selama timer 015 aktif, yaitu sekitar 130 detik (#1300). Pengaktifan sinyal pada proses ini dilakukan oleh register 15.07, sehingga port 100.02 aktif selama 1,1detik. Setelah timer 015 selesai melaksanakan instruksinya maka output kontak ladder NO TIM015 akan aktif sehingga proses dari ladder memasuki tahap keempat yaitu pengaktifan instruksi SBS 17, dan secara otomatis instruksi SBN 17 aktif, lalu memproses mesin 1 menjadi kondisi daur aktif 66,67% (C1). Mesin 1 pada daur aktif 66,67% diproses selama selama timer 017 aktif, yaitu sekitar 60 detik. Pengaktifan sinyal dilakukan oleh register 15.08, sehingga port 100.01 kembali aktif selama 1,1 detik. Setelah timer 017 selesai melaksanakan instruksinya maka output kontak ladder NO TIM017 akan aktif sehingga proses dari ladder memasuki tahap terakhir dari mesin 1 tipe A ini atau tahap kelima yaitu pengaktifan instruksi SBS 19, dan secara otomatis instruksi SBN 19 aktif, lalu memproses mesin 1 menjadi kondisi daur aktif 50%(C0). Mesin 1 pada tahap kelima ini diproses oleh timer 019 selama 10 detik. Berbeda dari tahap-tahap sebelumnya pada tahapan ini mesin berada pada tahapan delay pemberhentian dari proses. Setelah timer 019 selesai melaksanakan instruksinya maka output kontak ladder NO TIM019 akan aktif sehingga proses dari *etching PCB* oleh mesin 1 selesai.

Perhitungan waktu dari mulai tahapan pertama, kedua, ketiga, keempat, dan kelima berasal dari pembagian dari perkiraan waktu total selesainya proses *etching PCB*. Yang diperkirakan waktu total untuk tipe B adalah 320s. Nilai dari 320 detik adalah nilai yang didapatkan dari beberapa kali percobaan pada rangkaian yang akan diproses oleh mesin. Sehingga tahap pertama berlangsung selama 60 detik, tahap kedua berlangsung selama 60 detik, tahap ketiga adalah inti dari proses maka waktu ditambah dua kali menjadi 130 detik, tahap keempat kembali menjadi 60 detik, dimana pada tahap ini diperkirakan rangkaian PCB sudah selesai diproses dan pada tahap kelima adalah tahap pemberhentian mesin yang diberi delay 10 detik. Pemrograman sinyal failure terlihat pada ladder baris ketiga yang proses kerjanya adalah ketika input kontak 000.00 menjadi *Normally Closed*, maka output port pada PLC yaitu 100.04 akan aktif dan menghidupkan

relay K4 untuk membawa sinyal ke mikrokontroler AT89S51. Proses perubahan input kontak normally closed menjadi normally open dilakukan manual oleh saklar input pada mesin.

Proses pergerakan dari mesin 2 tipe A (fiber PCB) dijalankan oleh ladder bagian 5 dan ladder bagian 6 pada lampiran E dan F, yang mana pada bagian tersebut pengontrolan mesin 2 disertai dengan pemberian sinyal ke mikrokontroler AT89S51 dilakukan. Untuk pengontrolan mesin 2 ditandai dengan pemanggilan subroutine 2,4,6,8,10, yang dilakukan dengan instruksi ladder logic SBS(91). Adapun fungsi dari SBS(91) adalah memanggil subroutine yang ditandai dengan instruksi SBN(92), yang terletak pada ladder di lampiran J dan K. Pada bagian inilah PWM dengan daur aktif 50%, 66,67%, 83,33% di mesin 1 tipe A dijalankan. Ketika Tombol start M2 (000.04) di aktifkan maka register 12.01 aktif terkunci (latching) yang artinya adalah proses yang akan dijalankan pada register 12.01 tidak akan berhenti atau berulang, walaupun tombol 000.04 tertekan. Namun register 12.01 akan berhenti apabila tombol input 000.08 ditekan. Setelah register 12.01 aktif maka akan ada 3 proses yang dilakukan yaitu SBS 2 aktif, timer 002 aktif, dan register 16.00 aktif. Instruksi SBS 1 adalah pengaktifan instruksi SBN 2 yang mana didalam instruksi ini PWM mesin 2 pada kondisi daur aktif 50% di proses (C0). Mesin 2 pada daur aktif 50% ini diproses selama timer 002 aktif, yaitu sekitar 72 detik. Pada ladder 3.1.3 pada bagian 5 dituliskan pada intruksi timer 001 dengan cara #0720, yang berarti nilai timer pada omron 720 dibagi konstanta 10 yaitu 72 detik. Sedangkan pengaktifan register 016.00 mempunyai fungsi yang sudah dijelaskan pada tabel 3.1.2 yaitu pembawa sinyal ke mikrokontroler dengan cara mengaktifkan port 100.00 dan kemudian port 100.00 ini akan menhidupkan relay K1 aktif selama sekitar 0,8 detik (#0008). Setelah 0,8 detik maka output kontak ladder NO TIM40 akan aktif yang membuat output port 100.00 mati (reset). Dan ketika tim 002 selesai melaksanakan instruksinya maka input ladder NO TIM001 akan aktif sehingga proses dari ladder memasuki tahap kedua yaitu pengaktifan instruksi SBS 4. Dengan aktifnya SBS 4, maka instruksi SBN 4 akan memproses mesin 1 menjadi kondisi daur aktif 66,67% (C1). Mesin 2 pada daur aktif 66,67% ini diproses selama timer 004 aktif, yaitu sekitar 72 detik. Pengaktifan sinyal pada proses ini dilakukan oleh register 16.01, sehingga port 100.01

aktif, selama 0,6 detik. Setelah timer 004 selesai melaksanakan instruksinya maka output kontak ladder NO TIM004 akan aktif sehingga proses dari ladder memasuki tahap ketiga yaitu pengaktifan instruksi SBS 6. Sehingga instruksi SBN 6 aktif, lalu memproses mesin 2 menjadi kondisi daur aktif 83,33% (C2). Mesin 2 pada daur aktif 83,33% ini diproses selama timer 006 aktif, yaitu sekitar 134 detik (#1340). Pengaktifan sinyal pada proses ini dilakukan oleh register 16.02, sehingga port 100.02 aktif selama 0,8 detik. Setelah timer 006 selesai melaksanakan instruksinya maka output kontak ladder NO TIM006 akan aktif sehingga proses dari ladder memasuki tahap keempat yaitu pengaktifan instruksi SBS 8, dan secara otomatis instruksi SBN 8 aktif, lalu memproses mesin 2 menjadi kondisi daur aktif 66,67% (C1). Mesin 2 pada daur aktif 66,67% diproses selama selama timer 008 aktif, yaitu sekitar 72 detik. Pengaktifan sinyal dilakukan oleh register 16.01, sehingga port 100.01 kembali aktif selama 0,6 detik. Setelah timer 008 selesai melaksanakan instruksinya maka output kontak ladder NO TIM007 akan aktif sehingga proses dari ladder memasuki tahap terakhir dari mesin 2 tipe A ini atau tahap kelima yaitu pengaktifan instruksi SBS 10, dan secara otomatis instruksi SBN 10 aktif, lalu memproses mesin 2 menjadi kondisi daur aktif 50%(C0). Mesin 2 pada tahap kelima ini diproses oleh timer 010 selama 10 detik. Berbeda dari tahap-tahap sebelumnya pada tahapan ini mesin berada pada tahapan delay pemberhentian dari proses. Setelah timer 010 selesai melaksanakan instruksinya maka output kontak ladder NO TIM010 akan aktif sehingga proses dari *etching PCB* oleh mesin 1 selesai.

Perhitungan waktu dari mulai tahapan pertama, kedua, ketiga, keempat, dan kelima berasal dari pembagian dari perkiraan waktu total selesainya proses *etching PCB*. Yang diperkirakan waktu total untuk tipe A adalah 360s. Nilai dari 360 detik adalah nilai yang didapatkan dari beberapa kali percobaan pada rangkaian yang akan diproses oleh mesin. Sehingga tahap pertama berlangsung selama 72 detik, tahap kedua berlangsung selama 72 detik, tahap ketiga adalah inti dari proses maka waktu ditambah dua kali menjadi 114 detik, tahap keempat kembali menjadi 72 detik, dimana pada tahap ini diperkirakan rangkaian PCB sudah selesai diproses dan pada tahap kelima adalah tahap pemberhentian mesin yang diberi delay 10 detik.

Pemrograman sinyal failure terlihat pada ladder baris ketiga yang proses kerjanya adalah ketika input kontak 000.01 menjadi *Normally Closed*, maka output port pada PLC yaitu 100.06 akan aktif dan menhidupkan relay K4 untuk membawa sinyal ke mikrokontroller AT89S51. Proses perubahan input kontak normaly closed menjadi normally open dilakukan manual oleh saklar input pada mesin.

Proses pergerakan dari mesin 2 tipe B (Phenolic Cotton Paper) dijalankan oleh ladder bagian 7 dan ladder bagian 8 pada lampiran G dan H, yang mana pada bagian tersebut pengontrolan mesin 2 disertai dengan pemberian sinyal ke mikrokontroller AT89S51 dilakukan. Untuk pengontrolan mesin 1 ditandai dengan pemanggilan subroutine 12,14,16,18,20, yang juga dilakukan dengan instruksi ladder logic SBS(91). Adapun fungsi dari SBS(91) adalah memanggil subroutine yang ditandai dengan instruksi SBN(92), yang terletak pada ladder pada lampiran L dan M. Pada bagian inilah PWM dengan daur aktif 50%, 66,67%, 83,33% di mesin 2 tipe B dijalankan. Ketika Tombol start M2 (000.06) di aktifkan maka register 14.00 aktif terkunci (latching) yang artinya adalah proses yang akan dijalankan pada register 14.00 tidak akan berhenti atau berulang, walaupun tombol 000.05 tertekan. Namun register 14.00 akan berhenti apabila tombol input 000.08 ditekan. Setelah register 14.00 aktif maka akan ada 3 proses yang dilakukan yaitu SBS 12 aktif, timer 012 aktif, dan register 16.05 aktif. Instruksi SBS 12 adalah pengaktifan instruksi SBN 12 yang mana didalam instruksi ini PWM mesin 2 pada kondisi daur aktif 50% di proses (C0). Mesin 2 pada daur aktif 50 % ini diproses selama timer 012 aktif, yaitu sekitar 60 detik. Pada ladder 3.1.3 pada bagian 7 dituliskan pada intruksi timer 012 dengan cara #0600, yang berarti nilai timer pada omron 600 dibagi konstanta 10 yaitu 60 detik. Sedangkan pengaktifan register 016.05 mempunyai fungsi yang sudah dijelaskan pada tabel 3.1.2 yaitu pembawa sinyal ke mikrokontroller dengan cara mengaktifkan port 100.00 dan kemudian port 100.00 ini akan menhidupkan relay K1 aktif selama sekitar 0,6 detik (#0006). Setelah 1,1 detik maka output kontak ladder NO TIM45 akan aktif yang membuat output port 100.00 mati (reset). Dan ketika tim 012 selesai melaksanakan instruksinya maka input ladder NO TIM012 akan aktif sehingga proses dari ladder memasuki tahap kedua yaitu pengaktifan instruksi SBS 14. Dengan aktifnya SBS 14, maka instruksi SBN 14 akan memproses mesin 2 menjadi kondisi daur

aktif 66,67% (C1). Mesin 2 pada daur aktif 66,67% ini diproses selama timer 014 aktif, yaitu sekitar 60 detik. Pengaktifan sinyal pada proses ini dilakukan oleh register 16.06, sehingga port 100.01 aktif, selama 0,6 detik. Setelah timer 014 selesai melaksanakan instruksinya maka output kontak ladder NO TIM014 akan aktif sehingga proses dari ladder memasuki tahap ketiga yaitu pengaktifan instruksi SBS 16. Sehingga instruksi SBN 16 aktif, lalu memproses mesin 2 menjadi kondisi daur aktif 83,33% (C2). Mesin 2 pada daur aktif 83,33% ini diproses selama timer 016 aktif, yaitu sekitar 130 detik (#1300). Pengaktifan sinyal pada proses ini dilakukan oleh register 16.07, sehingga port 100.02 aktif selama 0,6 detik. Setelah timer 016 selesai melaksanakan instruksinya maka output kontak ladder NO TIM016 akan aktif sehingga proses dari ladder memasuki tahap keempat yaitu pengaktifan instruksi SBS 18, dan secara otomatis instruksi SBN 18 aktif, lalu memproses mesin 2 menjadi kondisi daur aktif 66,67% (C1). Mesin 2 pada daur aktif 66,67% diproses selama selama timer 018 aktif, yaitu sekitar 60 detik. Pengaktifan sinyal dilakukan oleh register 16.08, sehingga port 100.01 kembali aktif selama 0,6 detik. Setelah timer 018 selesai melaksanakan instruksinya maka output kontak ladder NO TIM018 akan aktif sehingga proses dari ladder memasuki tahap terakhir dari mesin 2 tipe A ini atau tahap kelima yaitu pengaktifan instruksi SBS 20, dan secara otomatis instruksi SBN 20 aktif, lalu memproses mesin 2 menjadi kondisi daur aktif 50%(C0). Mesin 2 pada tahap kelima ini diproses oleh timer 020 selama 10 detik. Berbeda dari tahap-tahap sebelumnya pada tahapan ini mesin berada pada tahapan delay pemberhentian dari proses. Setelah timer 020 selesai melaksanakan instruksinya maka output kontak ladder NO TIM020 akan aktif sehingga proses dari *etching PCB* oleh mesin 2 selesai.

Perhitungan waktu dari mulai tahapan pertama, kedua, ketiga, keempat, dan kelima berasal dari pembagian dari perkiraan waktu total selesainya proses *etching PCB*. Yang diperkirakan waktu total untuk tipe B adalah 320s. Nilai dari 320 detik adalah nilai yang didapatkan dari beberapa kali percobaan pada rangkaian yang akan diproses oleh mesin. Sehingga tahap pertama berlangsung selama 60 detik, tahap kedua berlangsung selama 60 detik, tahap ketiga adalah inti dari proses maka waktu ditambah dua kali menjadi 130 detik, tahap keempat kembali menjadi 60 detik, dimana pada tahap ini

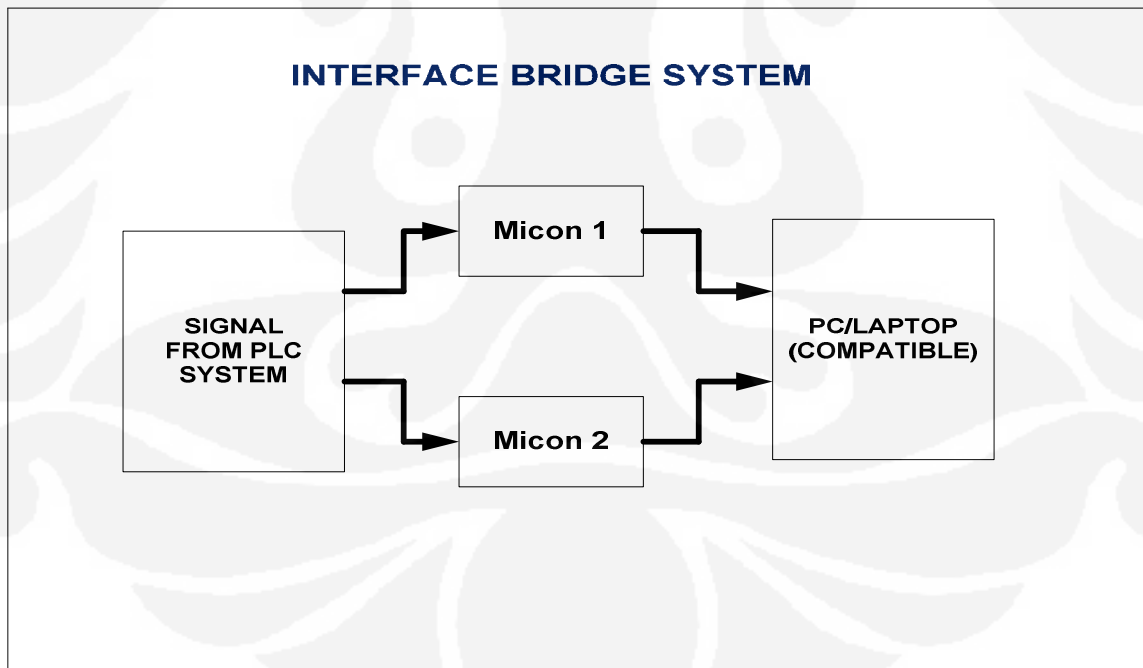
diperkirakan rangkaian PCB sudah selesai diproses dan pada tahap kelima adalah tahap pemberhentian mesin yang diberi delay 10 detik.

Pemrograman sinyal failure terlihat pada ladder baris ketiga yang proses kerjanya adalah ketika input kontak 000.01 menjadi *Normally Closed*, maka output port pada PLC yaitu 100.06 akan aktif dan menghidupkan relay K4 untuk membawa sinyal ke mikrokontroller AT89S51. Proses perubahan input kontak normaly closed menjadi normally open dilakukan manual oleh saklar input pada mesin.

3.3 Perancangan Sistem Bridge AT89S51

Sistem *bridge* ini merupakan bagian yang terpenting dalam sistem antarmuka keseluruhan. Sistem ini menggunakan 2 mikrokontroller sebagai penghubung ke port PC atau laptop yang sesuai (*Compatible*). Adapun fungsi dari mikrokontroller A untuk membawa sinyal dari mesin 1, sedangkan mikrokontroller B untuk membawa sinyal dari mesin 2 yang selanjutnya sinyal-sinyal tersebut diolah pada Visual basic 6.

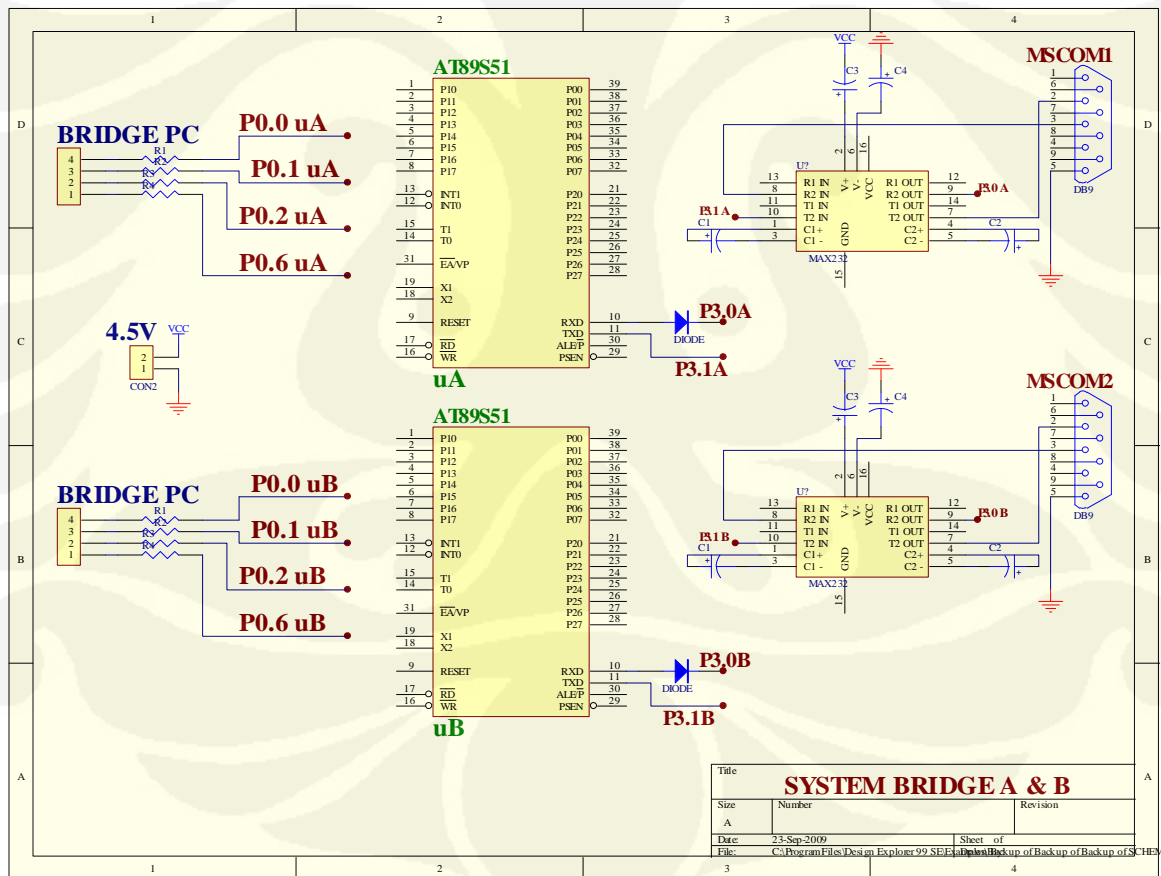
Pada blok diagram 3.2 dijelaskan bahwa mikrokontroller 1 dan 2 melewati 2 jalur input data ke PC.



Gambar 3.18 Blok Diagram Sistem Bridge

Seperti yang digambarkan pada gambar 3.19, yaitu sinyal-sinyal yang akan diolah pada VB6 ada 8. Untuk supply tegangan pada kedua mikrokontroller dibawah menggunakan satu sumber yaitu sekitar 4,5 V sampai 5 V dan arus 1000mA sampai 1200mA. Sedangkan pemrograman assembly mikrokontroller 1 dan 2 berdiri sendiri (*stand alone*) . Seperti penjelasan pada tabel 3.2 bahwa P0.0 merupakan sinyal kondisi 0 pada mesin (C0), P0.1 adalah sinyal kondisi 1 pada mesin (C1), P0.2 adalah sinyal kondisi 2 pada mesin (C2), dan P0.6 merupakan sinyal manual untuk menyatakan nilai failure pada proses. Keempat sinyal inilah yang akan dibawa ke mikrokontroller dan selanjutnya akan diproses oleh VB6. Oleh karena itu sistem ini dinamakan sistem Bridge.

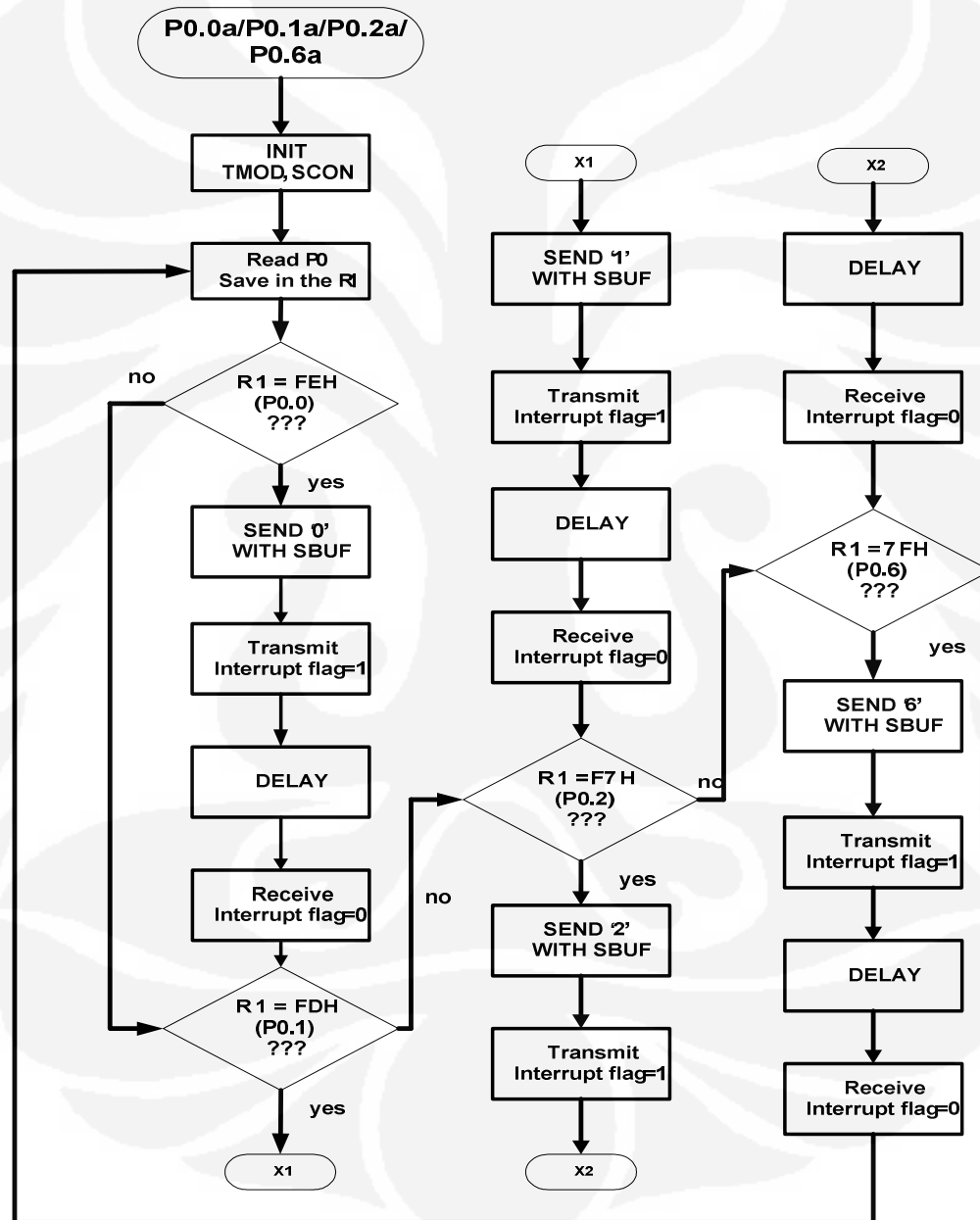
Adapun konsep dari pemrograman sistem ini adalah sederhana, yaitu mnegubah sinyal-sinyal dari PLC menjadi biner kemudian nilai dari bilangan biner tersebut akan dikirim melalui instruksi assembler, selanjutnya nilai-nilai biner tersebut akan ditampilkan oleh PC atau laptop yang sesuai dengan standar yang ditentukan.



Gambar 3.19 Skema Sistem *Bridge*

3.3.1 Perancangan Sistem Bridge A

Sistem *Bridge A* adalah sistem yang membawa sinyal-sinyal dari mesin 1 melalui mikrokontroller A. Tipe A dan Tipe B PCB pada mesin 1 melawati jembatan atau mikrokontroller yang sama. Flowchart 3.20 Bagian 3 merupakan gambaran cara kerja sistem Bridge A.



Gambar 3.20 Flowchart Bagian 3a

Gambar flowchart 3.20 ini merupakan lanjutan dari flowchart 3.15, yaitu pengolahan 4 sinyal yang dihasilkan dari proses mesin pada bagian selanjutnya. Dari tampilan flowchart tersebut dapat terlihat bahwa proses awal flowchart diawali dari masuknya salah satu sinyal ke port input mikrokontroler. Keempat port itu adalah P0.0, P0.1, P0.2, dan P0.6, yang mana huruf a menyatakan bahwa keempat sinyal tersebut akan dibawa melawati mikrokontroler A.

Gambar 3.21 merupakan hasil pemrograman oleh Reads51 yang menggunakan bahasa assembler.

<pre> #include <sfr51.inc> start: mov tmod,#20h mov scon,#01010000b mov th1,#-3 setb tr1 main: mov r1,p0 MOTORA: cjne r1,#11111110b, MOTORB mov a,#'0' mov sbuf,a jnb ti,\$ clr ti lcall delay ljmp start MOTORB: cjne r1,#111111101b, MOTORC mov a,#'1' mov sbuf,a jnb ti,\$ clr ti lcall delay ljmp start MOTORC: cjne r1,#11111011b, FAILURE mov a,#'2' mov sbuf,a jnb ti,\$ clr ti lcall delay ljmp start </pre>	<pre> FAILURE: cjne r1,#10111111b, jikax mov a,#'6' mov sbuf,a jnb ti,\$ clr ti lcall delay jikax: ljmp start receive: jnb ri,receive clr ri mov a,sbuf ret delay: mov r7,#8 delay1: mov r6,#0ffh delay2: mov r5,#00h djnz r5,\$ djnz r6,delay2 djnz r7,delay1 ret end </pre>
PAGE 1	PAGE 2

Gambar 3.21 Pemrograman Bridge A Dengan Reads 51

Setelah salah satu sinyal tersebut masuk ke mikrokontroller maka proses selanjutnya akan melalui 2 tahapan inisialisasi. Yang pertama adalah inisialisasi TMOD (#20h). Nilai dari TMOD pada Gambar 3.33 yang bernilai 20h dibuat menjadi biner adalah #0010000. Maka nilai biner ini apabila akan disesuaikan pada gambar 2.2.1 dan dimasukkan kedalam tabel 2.2.1.1 tentang register TMOD didapat data bahwa TMOD seperti ini mempunyai jenis Mode Operasi 2 yang mana pada mode ini baud rate dapat dihasilkan. Sesuai dengan instruksi pada source code, bahwa baud rate akan diberi nilai 9600 (mov th1,#-3). Dan inisialisasi dari SCON merupakan pembentukan nilai untuk mengaktifkan komunikasi serial. Berikutnya adalah penyimpanan nilai dari P0 yang masuk ke Register 1 (R1). Kemudian dilakukan pemeriksaan isi register apakah R1 sama dengan Feh (P0.0) ? jika benar maka kirim angka 0 melalui SBUF, bila telah selesai mengirimkan data maka TI (Transmit Interrupt flag) akan bernilai 1 dengan sendirinya selama *delay microcontroller* yang ditentukan atau telah diprogram oleh assembler. Tabel dibawah ini akan menjelaskan perhitungan waktu delay.

LISTING PROGRAM	PROCESS	CYCLE
delay : mov r7,# 8	1x	1
delay1: mov r6,#0ffh	1x	1
delay2: mov r5,#00h	1x	1
djnz r5,\$	256 X 255 X 8 = 522.240x	2
djnz r6,delay2	255	2
djnz r7,delay1	8x	2

Tabel 3.5 Perhitungan Delay Bridge A

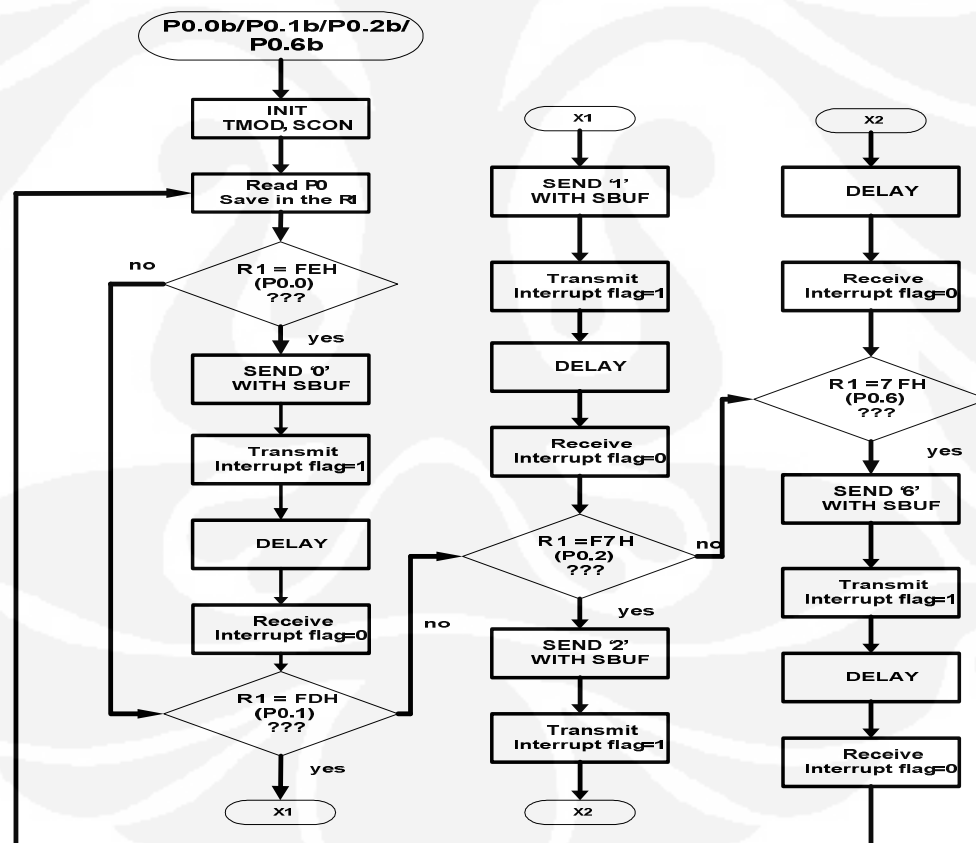
Baris 4 dikerjakan sebanyak 522.240x, karena instruksi tersebut dikerjakan selama 2 siklus maka proses totalnya menjadi $522.240 \times 2 = 1.044.480$ x. Sedangkan baris ke 5 dikerjakan 3 siklus sehingga menjadi 765 siklus (255×3), dan yang baris keenam menjadi 24 siklus (8×3). Sehingga total siklus adalah 1.045.293 siklus.

Untuk frekuensi kristal 12 MHz waktu yang dibutuhkan adalah 1.045.293 siklus dikalikan dengan 1µS maka hasil waktu yang dihasilkan adalah 1,045 detik. Nilai ini mendekati dengan delay yang dihasilkan pada PLC yang tergambar pada gambar ladder 3.1.3 bagian 1 pada timer 30 yaitu sebesar 1,1 detik.

Setelah delay melaksanakan kerja instruksinya maka tanda pemantau dari SBUF yang berupa RI (Receive Interrupt flag) akan menjadi 0. Pada source code 3.2.1 listing program berjudul receive yang akan membuat RI menjadi 0 sehingga akumulator a dapat dipakai menerima data lagi.

3.3.2 Perancangan Sistem Bridge B

Sistem *Bridge B* adalah sistem yang membawa sinyal-sinyal dari mesin 2 melalui mikrokontroller B. Tipe A dan Tipe B PCB pada mesin 2 melawati jembatan atau mikrokontroller yang sama.



Gambar 3.22 Flowchart Bagian 3b

Gambar flowchart 3.22 ini merupakan lanjutan dari flowchart 3.15, yaitu pengolahan 4 sinyal yang dihasilkan dari proses mesin pada bagian selanjutnya. Dari tampilan flowchart tersebut dapat terlihat bahwa proses awal flowchart diawali dari masuknya salah satu sinyal ke port input mikrokontroler. Keempat port itu adalah P0.0, P0.1, P0.2, dan P0.6, yang mana huruf a menyatakan bahwa keempat sinyal tersebut akan dibawa melawati mikrokontroler A.

Gambar 3.23 merupakan hasil pemrograman oleh Reads51 yang menggunakan bahasa assembler.

<pre> #include <sfr51.inc> start: mov tmod,#20h mov scon,#01010000b mov th1,#-3 setb tr1 main: mov r1,p0 MOTORA: cjne r1,#1111110b, MOTORB mov a,#'0' mov sbuf,a jnb ti,\$ clr ti delay: mov r7,#07h delay1: mov r6,#0ffh delay2: mov r5,#00h djnz r5,\$ djnz r6,delay2 djnz r7,delay1 ljmp start receive1: jnb ri,receive1 clr ri mov a,sbuf ret MOTORB: cjne r1,#1111101b, MOTORC mov a,#'1' mov sbuf,a jnb ti,\$ clr ti delay3: mov r7,#06h delay4: mov r6,#0ffh delay5: mov r5,#00h djnz r5,\$ djnz r6,delay5 djnz r7,delay4 ljmp start receive2: jnb ri,receive2 clr ri mov a,sbuf ret </pre> <p style="text-align: center;">PAGE 1</p>	<pre> MOTORC: cjne r1,#11111011b, FAILURE mov a,#'2' mov sbuf,a jnb ti,\$ clr ti delay6: mov r7,#06h delay7: mov r6,#0ffh delay8: mov r5,#00h djnz r5,\$ djnz r6,delay8 djnz r7,delay7 ljmp start receive3: jnb ri,receive3 clr ri mov a,sbuf ret FAILURE: cjne r1,#10111111b, jikax mov a,#'6' mov sbuf,a jnb ti,\$ clr ti delay9: mov r7,#08h delay10: mov r6,#0ffh delay11: mov r5,#00h djnz r5,\$ djnz r6,delay11 djnz r7,delay10 ljmp start jikax: ljmp start receive4: jnb ri,receive4 clr ri mov a,sbuf ret end </pre> <p style="text-align: center;">PAGE 2</p>
--	--

Gambar 3.23 Pemrograman Bridge B Dengan Reads 51

Setelah salah satu sinyal tersebut masuk ke mikrokontroller maka proses selanjutnya akan melalui 2 tahapan inisialisasi. Yang pertama adalah inisialisasi TMOD (#20h). Nilai dari TMOD pada gambar 3.35 yang bernilai 20h dibuat menjadi biner adalah #0010000. Maka nilai biner ini apabila akan disesuaikan pada gambar 2.2.1 dan dimasukkan kedalam tabel 2.2.1.1 tentang register TMOD didapat data bahwa TMOD seperti ini mempunyai jenis Mode Operasi 2 yang mana pada mode ini baud rate dapat dihasilkan. Sesuai dengan instruksi pada source code, bahwa baud rate akan diberi nilai 9600 (mov th1,#-3). Dan inisialisasi dari SCON merupakan pembentukan nilai untuk mengaktifkan komunikasi serial. Berikutnya adalah penyimpanan nilai dari P0 yang masuk ke Register 1 (R1). Kemudian dilakukan pemeriksaan isi register apakah R1 sama dengan Feh (P0.0) ? jika benar maka kirim angka 0 melalui SBUF, bila telah selesai mengirimkan data maka TI (Transmit Interrupt flag) akan bernilai 1 dengan sendirinya selama *delay microcontroller* yang ditentukan atau telah diprogram oleh assembler. Tabel dibawah ini akan menjelaskan perhitungan waktu delay.

LISTING PROGRAM	PROCESS	CYCLE
delay: mov r7,#7	1x	1
delay1: mov r6,#0ffh	1x	1
delay2: mov r5,#00h	1x	1
djnz r5,\$	256 X 255 X 7 =456.960x	2
djnz r6,delay2	255	2
djnz r7,delay1	7x	2

Tabel 3.6 Perhitungan Delay Bridge B

Baris 4 dikerjakan sebanyak 456.960x, karena instruksi tersebut dikerjakan selama 2 siklus maka proses totalnya menjadi $456.960 \times 2 = 913.920x$. Sedangkan baris ke 5 dikerjakan 3 siklus sehingga menjadi 765 siklus (255×3), dan yang baris keenam menjadi 21 siklus (7×3). Sehingga total siklus adalah 914.706 siklus.

Untuk frekuensi kristal 12 MHz waktu yang dibutuhkan adalah 914.706 siklus dikalikan dengan $1\mu\text{s}$ maka hasil waktu yang dihasilkan adalah 0,9 detik. Nilai ini mendekati dengan delay yang dihasilkan pada PLC yang tergambar pada gambar ladder 3.1.3 bagian 1 pada timer 30 yaitu sebesar 1,1 detik.

Nilai dari sistem bridge B ini berbeda dengan sistem bridge A. Pada sistem ini nilai delay bervariasi yang terlihat pada source code 3.2.2. hal ini disebabkan oleh pengaruh daya pada sumber. Sehingga nilai dari delay harus diatur agar tidak terjadi kesalahan dalam pengiriman data.

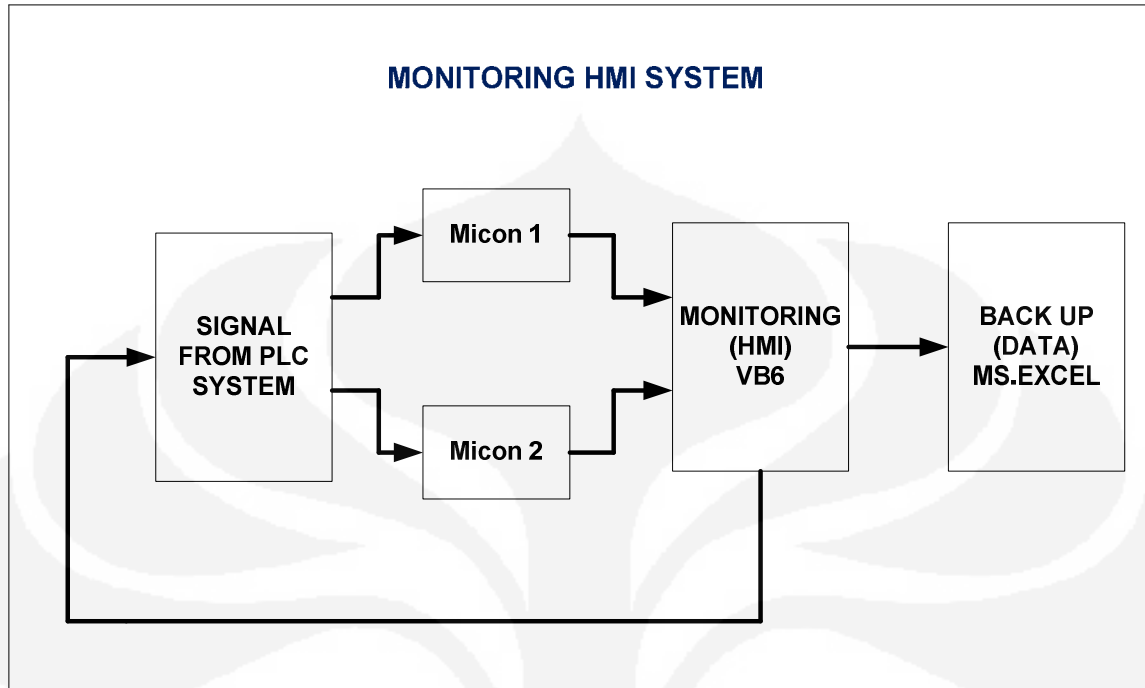
Setelah delay melaksanakan kerja instruksinya maka tanda pemantau dari SBUF yang berupa RI (Receive Interrupt flag) akan menjadi 0. Pada source code 3.2.1 listing program berjudul receive yang akan membuat RI menjadi 0 sehingga akumulator a dapat dipakai menerima data lagi.

3.4 Perancangan Sistem Monitoring HMI dengan VB6

Konsep dari sistem Monitoring HMI dengan VB6 ini adalah dengan mengambil sinyal – sinyal yang dihasilkan oleh PLC dan sinyal tersebut dibawa oleh Mikrokontroler lalu sinyal-sinyal tersebut diolah oleh VB6 dan selanjutnya ditampilkan oleh PC atau laptop yang *compatible*.

Adapun Penempatan dari Monitoring ini lebih diutamakan di ruangan bagian produksi. Sehingga konsep dari tampilan HMI adalah tampilan dari perhitungan hasil produksi dimulai dari hari Senin sampai Jumat, dan untuk hari Sabtu dan Minggu dianggap bukan jam kerja produksi. Namun tidak terlepas dari konsep tampilan yang akan dibuat, HMI ini diharapkan mampu memberikan sebagian informasi-informasi keteknisan yang sangat berguna sekali bagi pengembangan-pengembangan produksi nantinya. Seperti Time proses, daur aktif. Hal ini dikarenakan yang seharusnya melakukan pengembangan-pengembangan sistem produksi adalah bagian produksi yang mengerti tentang ilmu teknik didukung oleh bagian *maintenance*.

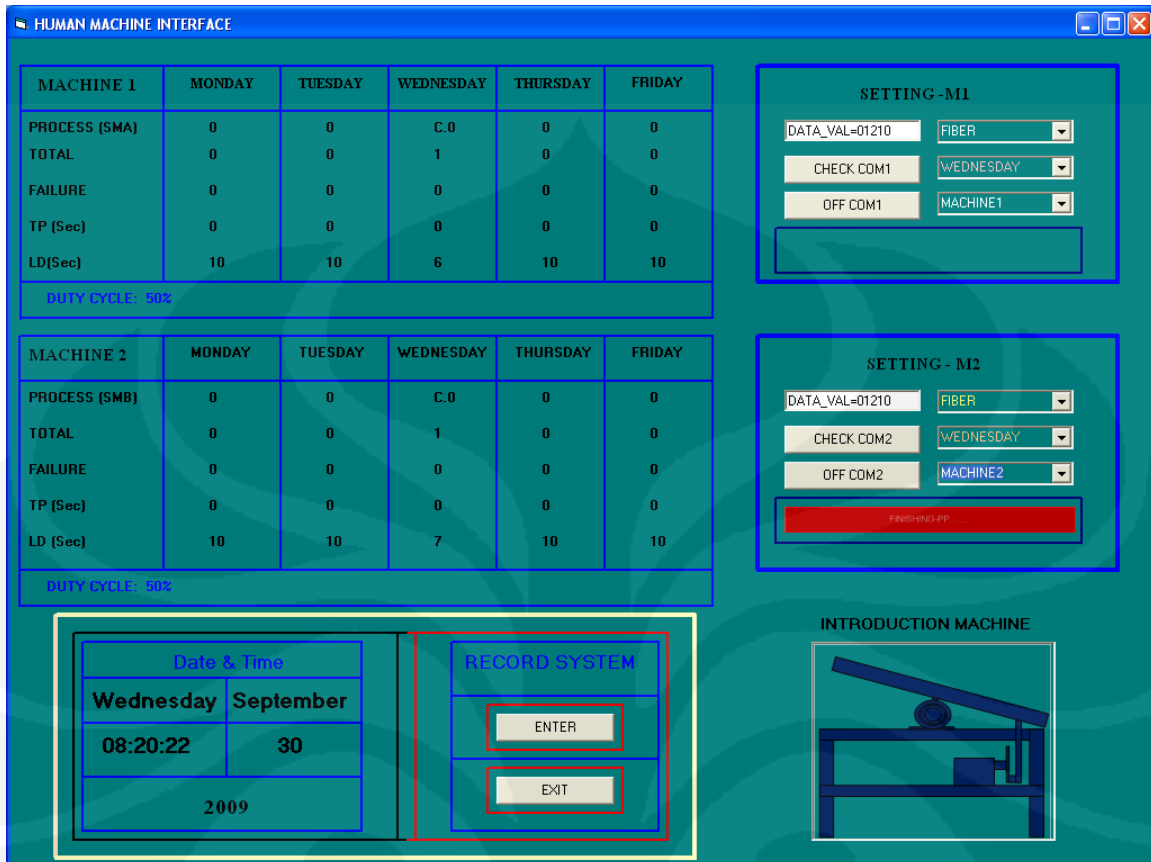
Pada blok diagram 3.34 dapat dijelaskan bahwa VB6 akan mengolah sinyal-sinyal dari VB6, selanjutnya hasil data hasil pengolahan tersebut akan dikirim ke MS.Excel sebagai *back up*.



Gambar 3.24 Blok Diagram Sistem Monitoring

Monitoring HMI dalam sistem ini mempunyai 2 fungsi yaitu membaca data dari mesin (*read*) dan juga dapat menulis instruksi kepada mesin (*write*) sehingga data yang didapat sesuai dengan apa yang diinstruksikan oleh PC. Misalnya data yang ingin ditampilkan mesin 1 maka pada monitoring HMI hanya menampilkan data pada tabel Machine 1, Apabila data yang ingin ditampilkan dari mesin 2 maka monitoring HMI hanya menampilkan data pada tabel Machine 2. Sedangkan apabila data yang ingin ditampilkan mesin1 dan mesin 2 maka monitoring HMI dapat menampilkan kedua data tersebut [ada PC atau laptop yang *compatible*].

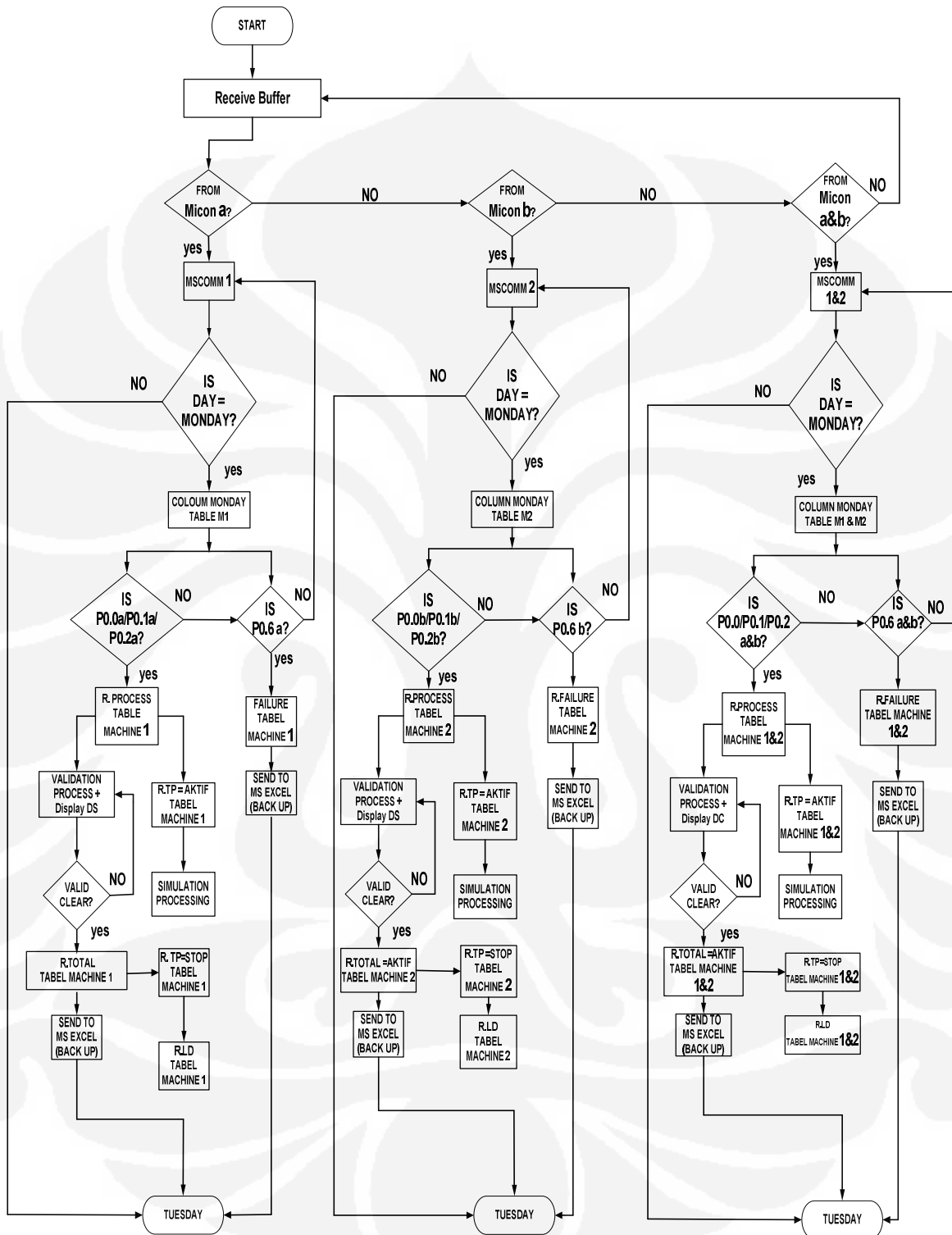
Gambar 3.25 merupakan tampilan monitoring HMI yang tampilannya terdiri dari Tabel Machine 1, Tabel Machine 2, Sistem Record, Date & Time, Introduction Machine, dan Box Setting untuk mesin 1 dan mesin 2 pada kondisi *RUN*. Pada Flowchart 3.1.3 bagian 4 adalah sistem kerja dari sinyal AT89s51 yang diolah ke VB6.



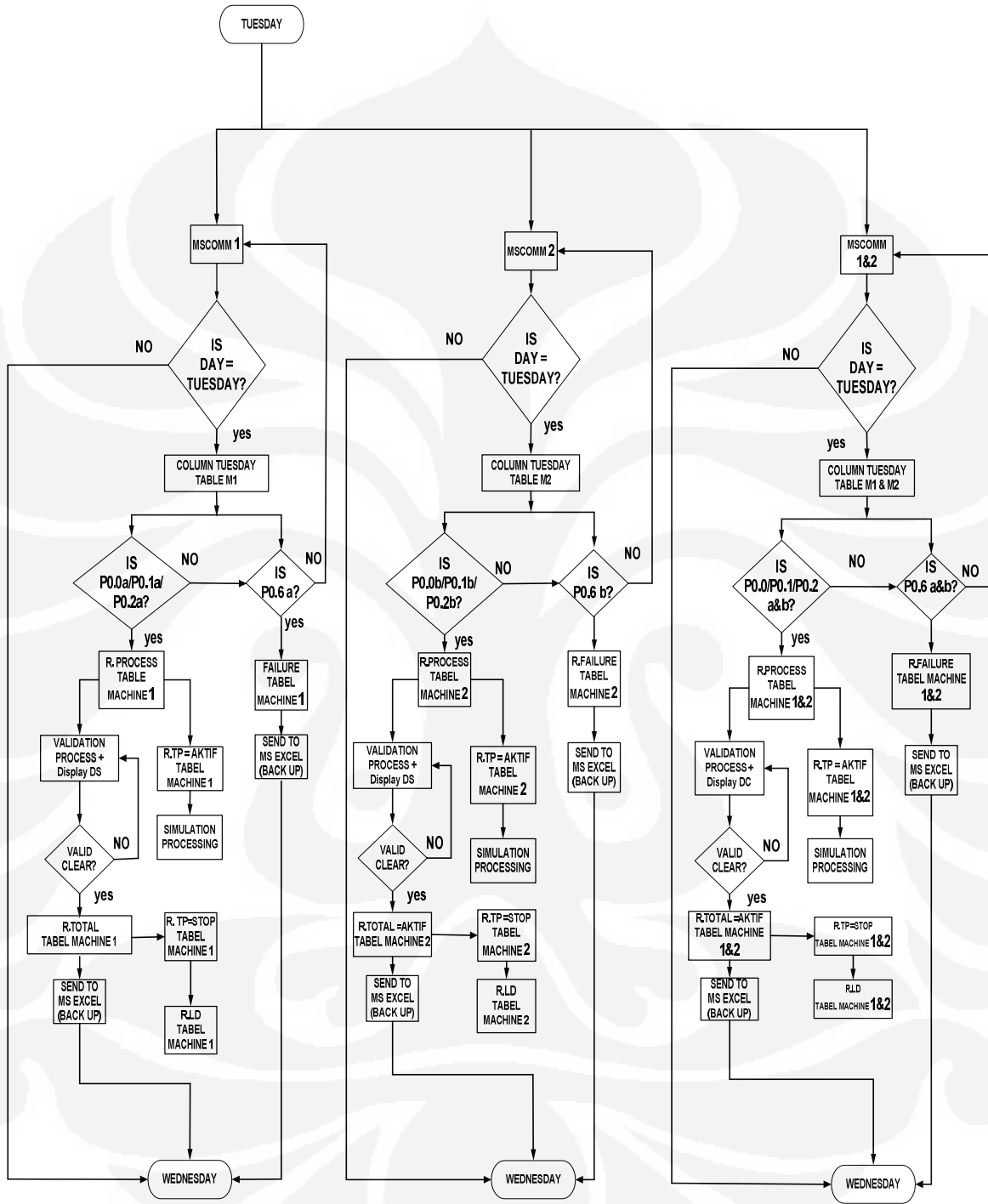
Gambar 3.25 Monitoring HMI (Run)

Tabel Machine1 terdiri dari tampilan interface untuk Process Signal Microcontroler A (SMA), Total, Failure, TP (Time Process), Last Delay, Duty Cycle. Begitu pula dengan Tabel Machine 2. Pada kotak setting-M1 dan setting-M2 mempunyai 7 bagian. Bagian pertama adalah text box yang bertuliskan DATA_VAL = 01210, artinya adalah proses etsa PCB sudah melawati proses C0,C1,C2,C1,CO (5 tahapan). Setiap tahapan-tahapan pada process akan masuk ke validasi 0,1,2,1,0 yang nantinya tahapan dari validasi ini digunakan untuk menghasilkan nilai TOTAL hasil PCB yang telah selesai di etsa.

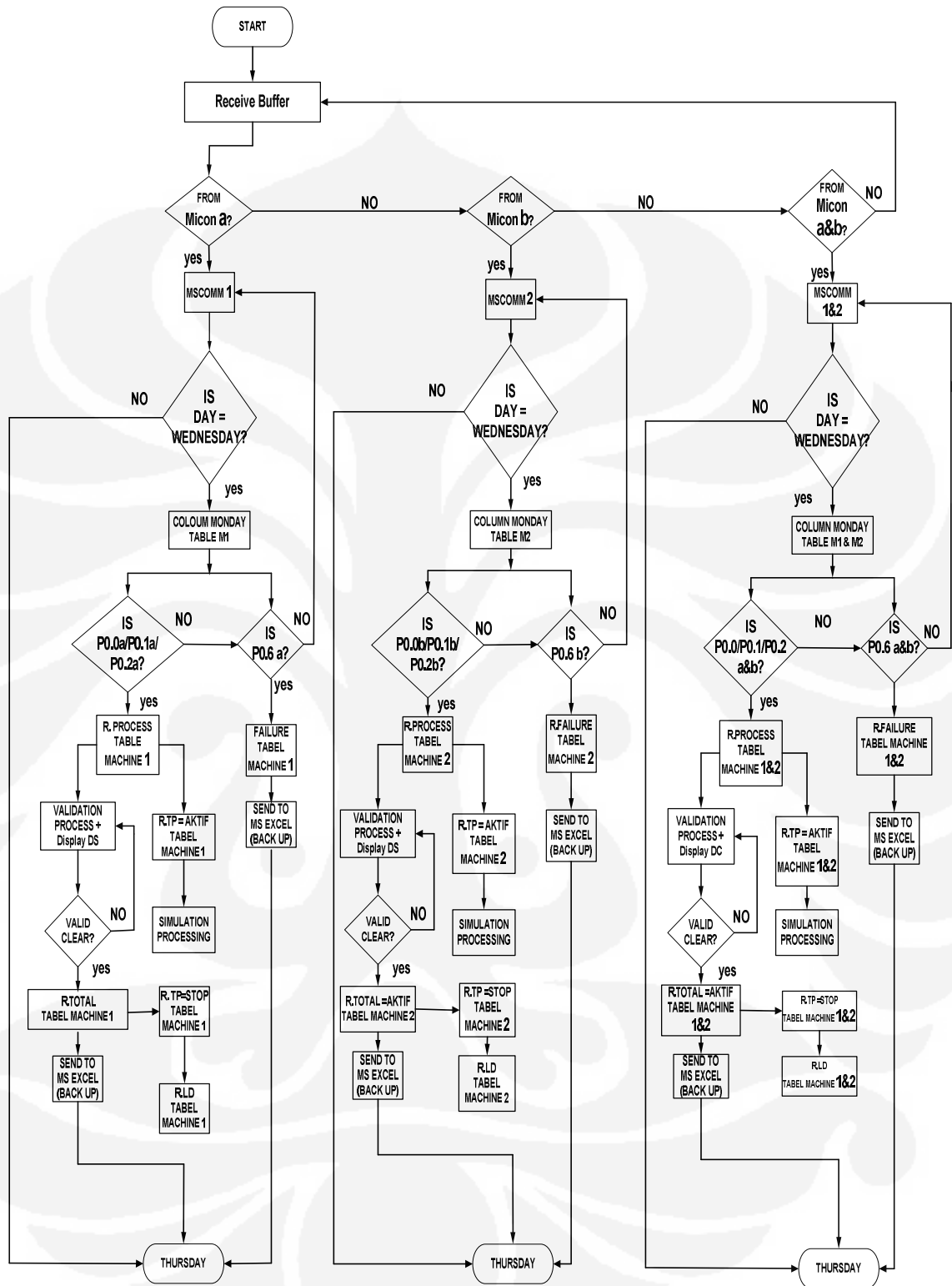
Untuk tombol CHECK COM1 merupakan tombol yang digunakan untuk mengaktifkan port mscomm1, dan sebaliknya tombol OFF COM1 digunakan untuk mematikan port mscomm1, Berikutnya adalah 3 list box yang fungsinya memilih type CB, Hari, dan penegasan mesin yang dipakai. Dan kotak terakhir dibawah merupakan tampilan gambar yang akan menyatakan proses berlangsung.



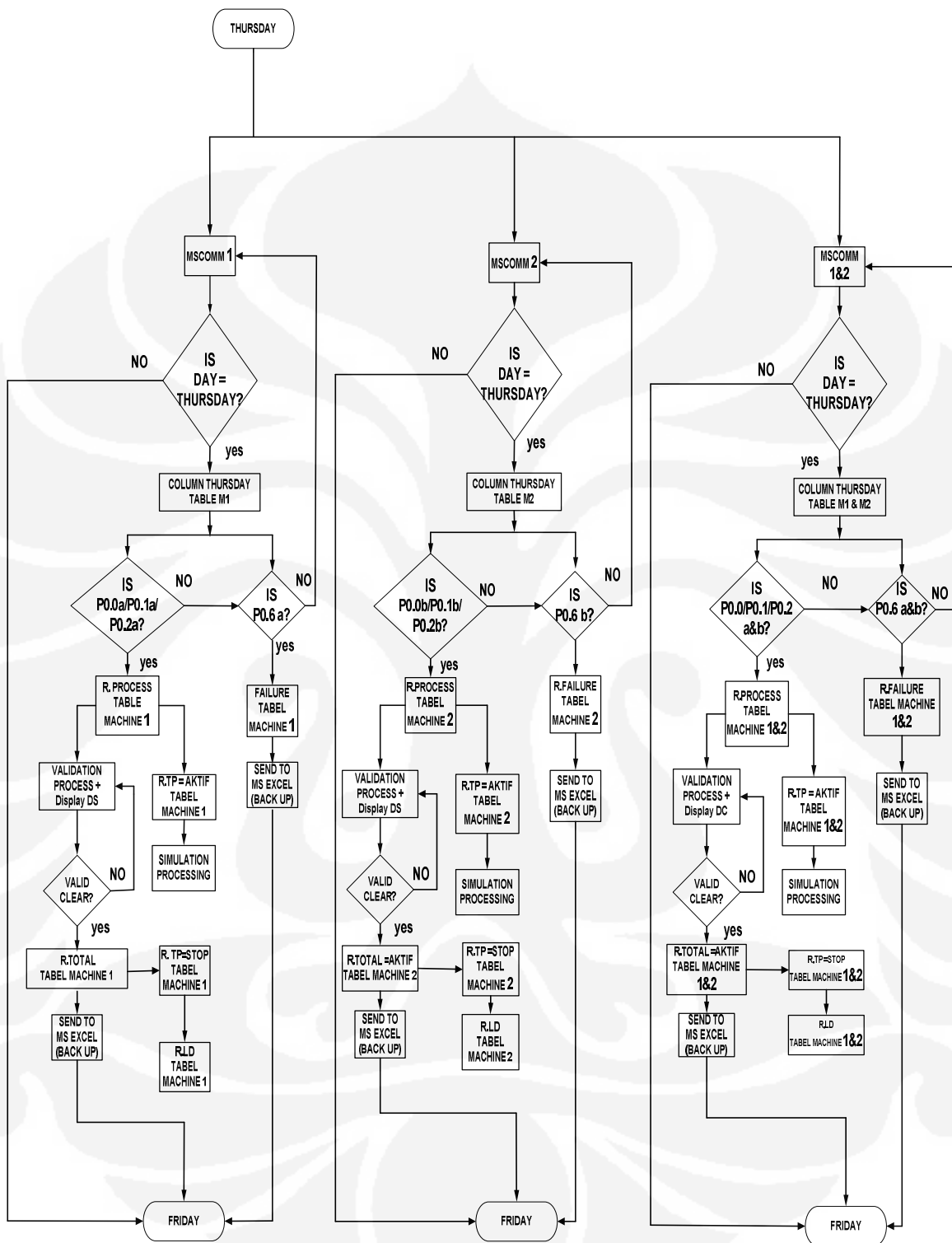
Gambar 3.26 Flowchart Bagian 4



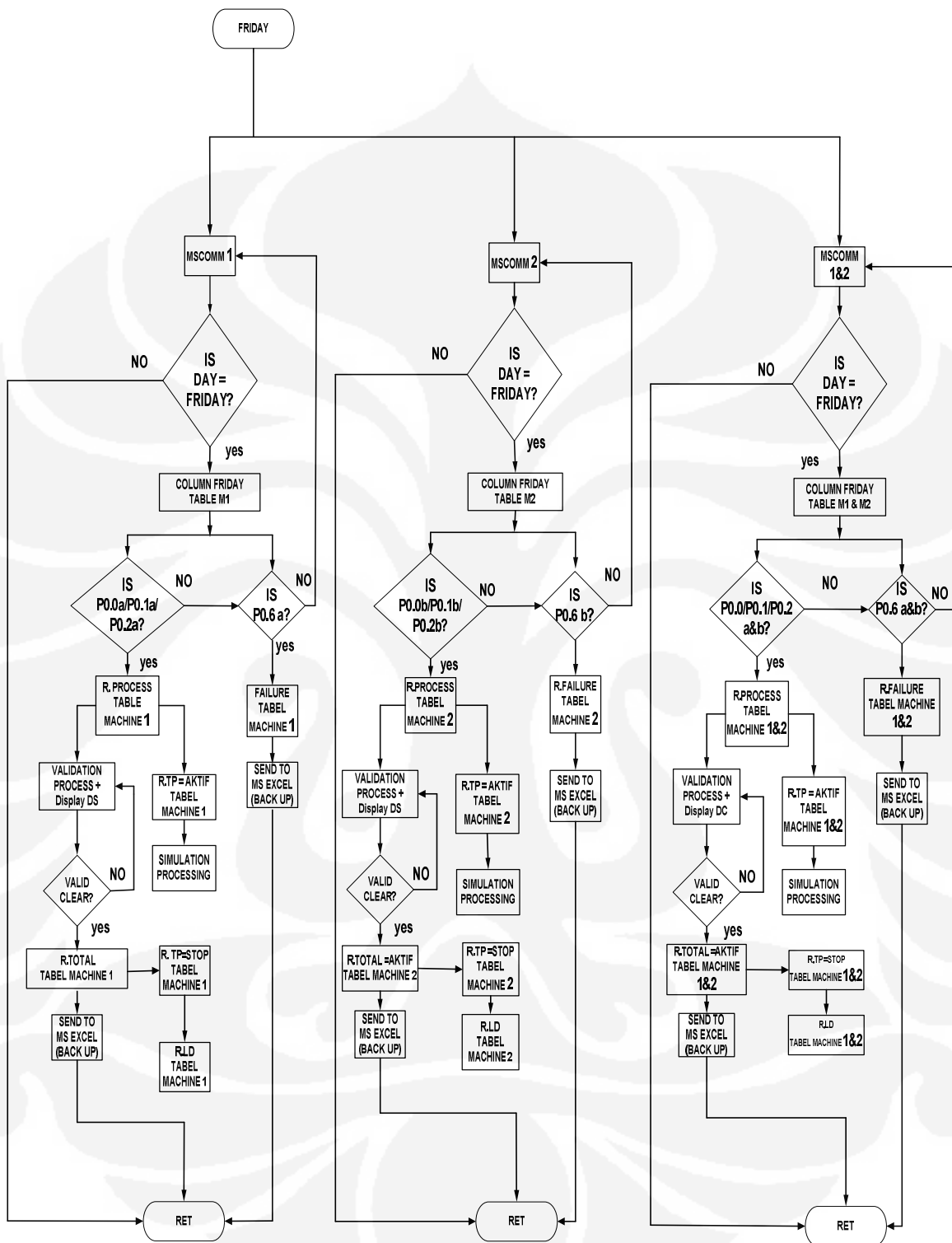
Gambar 3.27 Flowchart Bagian 5



Gambar 3.28 Flowchart Bagian 6



Gambar 3.29 Flowchart Bagian 7



Gambar 3.30 Flowchart Bagian 8

Seperti yang tergambar pada Flowchart 3.36 bagian 4 bahwa data disalin dari SBUF melalui penyangga penerima (*receive buffer*). Setelah itu data tersebut akan melawati tahap seleksi dari VB6, yang terdiri dari data dari mikrokontroler a, mikrokontroler b, atau sinyal yang terkirim secara bersamaan dari mikrokontroler a dan mikrokontroler b.

Data dari mikrokontroler a masuk melalui jalur mscomm 1 yang berasal dari comm port 3 pada PC atau laptop yang akan dipakai nantinya, sedangkan mikrokontroler b akan masuk melalui jalur mscomm 2 yang berasal dari comm port 1 pada PC atau laptop. Adapun mscomm 1 akan mengolah data dari mesin 1 dan mscomm 2 akan mengolah data dari mesin 2. List code dibawah ini merupakan proses dari mscomm 1 dan mscomm2.

```

Private Sub MSComm1_OnComm()
Dim inp As Integer
Select Case MSComm1.CommEvent
Case comEvReceive
If MSComm1.InBufferCount >= 0 Then
BUFFER = CStr(MSComm1.Input)
Text1.Text = BUFFER

inp = Val(BUFFER)
If inp > 7 Then 'ada 8 bit
inp = 0
Else

DATA_IN = DATA_IN & BUFFER
Text1.Text = "DATA_VAL=" & DATA_IN
-----
If MACHINE = "MACHINE1" Then

Private Sub MSComm2_OnComm()
Dim inpB As Integer
Select Case MSComm2.CommEvent
Case comEvReceive
If MSComm2.InBufferCount >= 0 Then
BUFFER2 = CStr(MSComm2.Input)
Text2.Text = BUFFER2

inpB = Val(BUFFER2)
If inpB > 7 Then 'ada 8 bit
inpB = 0
Else

DATA_IN2 = DATA_IN2 & BUFFER2
Text2.Text = "DATA_VAL=" & DATA_IN2
-----
If MACHINEB = "MACHINE2" Then

```

Gambar 3.31 Pemrograman VB6 Main 1

Pada gambar potongan List code 3.3 Main 1 diatas yang merupakan bagian terpenting dari pemrograman Visual Basic 6 sebagai pengolahan sinyal-sinyal yang masuk dari mikrokontroller. Pada bagian program inilah kunci pemrograman dari jalur data yang akan masuk dari port 3 (mscomm1) dan port 1 (mscomm2).

Bagian program main 1 menjelaskan bahwa data yang dapat dapat diterima pada mscomm berjumlah 8 (dari 0 sampai 7) , namun pada sistem ini tiap mscomm hanya mengolah 4 sinyal, yaitu 0,1,2,sebagai proses dan 6 sebagai failure.

Pengolahan sinyal yang pertama adalah melakukan seleksi hari. Ketika seleksi hari ditentukan, misalnya pada hari senin. Maka proses penulisan data pada monitoring HMI berada pada deret proses (*Row Process*) yang berada sejajar dengan kolom yang berjudul senin (*Column Monday*). Nilai proses yang ditampilkan berupa C0,C1,C2,C1,C0, dan akan selalu berubah sesuai dengan perubahan kondisi mesin. Yang mana setiap dari nilai proses ini akan selalu diolah dalam proses Validasi. Untuk mesin 1 tahapan validasi 11,12,13,14,dan 15. sedangkan untuk mesin 2 terdiri dari tahapan validasi 21,22,23,24,dan 25. Penulisan program pada bagian ini terlihat pada list code 3.4 main 2.

<pre> ----- If MACHINE = "MACHINE1" Then If DAYS = "MONDAY" Then LPMONDAY.Caption = "C." & BUFFER Timer1.Enabled = 1 ElseIf DAYS = "TUESDAY" Then LPTUESDAY.Caption = "C." & BUFFER Timer2.Enabled = 1 ElseIf DAYS = "WEDNESDAY" Then LPWEDNESDAY.Caption = "C." & BUFFER Timer3.Enabled = 1 ElseIf DAYS = "THURSDAY" Then LPTHURSDAY.Caption = "C." & BUFFER Timer4.Enabled = 1 ElseIf DAYS = "FRIDAY" Then LPFRIDAY.Caption = "C." & BUFFER Timer5.Enabled = 1 End If If Len(DATA_IN) = 1 Then Call VALIDASI11 ElseIf Len(DATA_IN) = 2 Then Call VALIDASI12 ElseIf Len(DATA_IN) = 3 Then Call VALIDASI13 ElseIf Len(DATA_IN) = 4 Then Call VALIDASI14 ElseIf Len(DATA_IN) = 5 Then Call VALIDASI15 End If </pre>	<pre> ----- If MACHINEB = "MACHINE2" Then If DAYS = "MONDAY" Then MPMONDAY.Caption = "C." & BUFFER2 Timer6.Enabled = 1 ElseIf DAYS = "TUESDAY" Then MPTUESDAY.Caption = "C." & BUFFER2 Timer7.Enabled = 1 ElseIf DAYS = "WEDNESDAY" Then MPWEDNESDAY.Caption = "C." & BUFFER2 Timer8.Enabled = 1 ElseIf DAYS = "THURSDAY" Then MPTHURSDAY.Caption = "C." & BUFFER2 Timer9.Enabled = 1 ElseIf DAYS = "FRIDAY" Then MPFRIDAY.Caption = "C." & BUFFER2 Timer10.Enabled = 1 End If If Len(DATA_IN2) = 1 Then Call VALIDASI21 ElseIf Len(DATA_IN2) = 2 Then Call VALIDASI22 ElseIf Len(DATA_IN2) = 3 Then Call VALIDASI23 ElseIf Len(DATA_IN2) = 4 Then Call VALIDASI24 ElseIf Len(DATA_IN2) = 5 Then Call VALIDASI25 End If </pre>
---	---

Gambar 3.32 Pemrograman VB6 Main 2

Validasi 11 pada list code 3.4 Main 2 merupakan penerimaan sinyal '0' dari mikrokontroller a yang tergambar pada Flowchart 3.32 bagian 3a (SEND '0' WITH SBUF), validasi 12 merupakan penerimaan sinyal '1', validasi 13 penerimaan sinyal '2', validasi 14 juga merupakan penerimaan sinyal '1', dan validasi 15 merupakan penerimaan sinyal '0'. Proses ini dapat dilihat pada SubVALIDASI di Source Code 3.3 main 3. Hasil dari validasi ini juga akan diproses untuk mendapatkan nilai total PCB, yang dilakukan oleh instruksi Len(DATA_IN) yang berfungsi mengumpulkan 5 proses validasi, yaitu validasi 11 sampai 15 ataupun validasi 21 sampai 25. Sedangkan fungsi Timer 1 sampai Timer 10 adalah berguna untuk untuk mengaktifkan Deret waktu proses (*Row Time Process*).

<pre> Sub VALIDASI11() If Left(DATA_IN, 1) = "0" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE0 Call simulation End Sub Sub VALIDASI12() If Mid(DATA_IN, 2, 1) = "1" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE1 Call simulation End Sub Sub VALIDASI13() If Mid(DATA_IN, 3, 1) = "2" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE2 Call simulation1 End Sub Sub VALIDASI14() If Mid(DATA_IN, 4, 1) = "1" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE1 Call simulation1 End Sub Sub VALIDASI15() If Left(DATA_IN, 1) = "0" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE0 Timer1.Enabled = True Call simulation2 End Sub Sub VALIDASI16() If Val(DATA_IN, 6) = "6" Then BENAR = True Else BENAR = False End If Call VALIDASISALAH End Sub </pre>	<pre> Sub VALIDASI21() If Left(DATA_IN2, 1) = "0" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call simulationa Call DUTYCYCLEA End Sub Sub VALIDASI22() If Mid(DATA_IN2, 2, 1) = "1" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call DUTYCYCLEB Call simulationa End Sub Sub VALIDASI23() If Mid(DATA_IN2, 3, 1) = "2" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call simulationb Call DUTYCYCLEC End Sub Sub VALIDASI24() If Mid(DATA_IN2, 4, 1) = "1" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call simulationb Call DUTYCYCLEB End Sub Sub VALIDASI25() If Left(DATA_IN2, 1) = "0" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call simulationc Call DUTYCYCLEA Timer12.Enabled = True End Sub Sub VALIDASI26() If Val(DATA_IN, 6) = "6" Then BENAR = True Else BENAR = False End If Call VALIDASISALAH2 End Sub </pre>
--	--

Gambar 3.33 Pemrograman VB6 Main 3

Seperti yang tertulis pada list source code 3.5 main 3 bahwa setiap tahapan proses validasi akan dikirim ke sub programnya (VALIDASIBENAR). Dan pada bagian sub program inilah nilai total dapat diproses sesuai pada Source Code 3.3 main 4. Selain untuk mencari nilai pada deret total (*Row Total*), pada program main 3 ini juga membuat 2 instruksi lainnya, yaitu menampilkan nilai dari *Duty Cycle* pada monitoring HMI, sehingga nilai yang tampil pada layar monitor sesuai dengan kondisi mesin. Berikutnya adalah menampilkan *Simulation Processing* yang berfungsi sebagai indikator proses etsa PCB yang lagi berjalan dan sudah selesai. Untuk sinyal failure ada pada validasi 16 untuk mesin1 dan validasi 26 untuk mesin 2. Sinyal ini bekerja secara manual (dijalankan operator) dan tidak masuk ke dalam sistem pada source code Main 2.

<pre> Sub VALIDASIBENAR() If MACHINE = "MACHINE1" Then If BENAR = True Then If Len(DATA_IN) >= 5 Then If DAYS = "MONDAY" Then LTMONDAY.Caption = Val(LTMONDAY.Caption) + 1 Call TimerA ElseIf DAYS = "TUESDAY" Then LTTUESDAY.Caption = Val(LTTUESDAY.Caption) + 1 Call TIMERB ElseIf DAYS = "WEDNESDAY" Then LTWEDNESDAY.Caption = Val(LTWEDNESDAY.Caption) + 1 Call TIMERC ElseIf DAYS = "THURSDAY" Then LTTHURSDAY.Caption = Val(LTTHURSDAY.Caption) + 1 Call TIMERD ElseIf DAYS = "FRIDAY" Then LTFRIDAY.Caption = Val(LTFRIDAY.Caption) + 1 Call TIMERE End If DATA_IN = "" End If </pre>	<pre> Sub VALIDASIBENAR2() If MACHINEB = "MACHINE2" Then If BENAR2 = True Then If Len(DATA_IN2) >= 5 Then If DAYS = "MONDAY" Then MTMONDAY.Caption = Val(MTMONDAY.Caption) + 1 Call TIMERF ElseIf DAYS = "TUESDAY" Then MTTUESDAY.Caption = Val(MTTUESDAY.Caption) + 1 Call TIMERG ElseIf DAYS = "WEDNESDAY" Then MTWEDNESDAY.Caption = Val(MTWEDNESDAY.Caption) + 1 Call TIMERH ElseIf DAYS = "THURSDAY" Then MTTHURSDAY.Caption = Val(MTTHURSDAY.Caption) + 1 Call TIMERI ElseIf DAYS = "FRIDAY" Then MTFRIDAY.Caption = Val(MTFRIDAY.Caption) + 1 Call TIMERJ End If DATA_IN2 = "" End If </pre>
--	---

Gambar 3.34 Pemrograman VB6 Main 4

Pada bagian inilah nilai total diproses sekaligus ditampilkan pada layar monitor HMI, melalui instruksi `Len(DATA_IN) >= 5` maka nilai Total dapat ditampilkan. Adapun maksud dari instruksi itu adalah ketika jumlah data yang masuk mencapai dengan 5 tahapan maka nilai dari total akan ditampilkan dan terus berulang apabila proses berlanjut (`Val(LTMONDAY.Caption) + 1`). Dan seterusnya akan seperti itu pada hari-hari selanjutnya.

Selain menampilkan nilai total pemrograman pada Main 4 ini juga akan memanggil timer A sampai J yang fungsinya adalah mengaktifkan sub program untuk *Last Delay* dan pengaktifan sitem *Record* , sehingga data nilai total dan failure bisa dicatat ke MS.Excel yang *compatible*.

<pre> Private Sub TimerA() LTPMONDAY.Caption = 0 Timer1.Enabled = 0 TimerLD1 = True TimerLD1X = True Call DATAEXPORT End Sub Private Sub TIMERB() LTPTUESDAY.Caption = 0 Timer2.Enabled = 0 TimerLD2 = True TimerLD2X = True Call DATAEXPORT1 End Sub Private Sub TIMERC() LTPWEDNESDAY.Caption = 0 Timer3.Enabled = 0 TimerLD3 = True TimerLD3X = True Call DATAEXPORT2 End Sub Private Sub TIMERD() LTPTHURSDAY.Caption = 0 Timer4.Enabled = 0 TimerLD4 = True TimerLD4X = True Call DATAEXPORT3 End Sub Private Sub TIMERE() LTPFRIDAY.Caption = 0 Timer5.Enabled = 0 TimerLD5 = True TimerLD5X = True Call DATAEXPORT4 End Sub </pre>	<pre> Private Sub TIMERF() MTPMONDAY.Caption = 0 Timer6.Enabled = False TimerLDA = True TimerLDAX = True DATAEXPORTA End Sub Private Sub TIMERG() MTPTUESDAY.Caption = 0 Timer7.Enabled = False TimerLDB = True TimerLDBX = True DATAEXPORTB End Sub Private Sub TIMERH() MTPWEDNESDAY.Caption = 0 Timer8.Enabled = False TimerLDC = True TimerLDCX = True DATAEXPORTC End Sub Private Sub TIMERI() MTPTHURSDAY.Caption = 0 Timer9.Enabled = False TimerLDD = True TimerLDDX = True DATAEXPORTD End Sub Private Sub TIMERJ() MTPFRIDAY.Caption = 0 Timer10.Enabled = False TimerLDE = True TimerLDEX = True DATAEXPORTE End Sub </pre>
---	--

Gambar 3.35 Pemrograman VB6 Main 5

TimerLD merupakan instruksi utama yang mengaktifkan deret delay terakhir (*Row LD*) pada tabel mesin 1 dan 2. Sedangkan instruksi inti dari sistem *record* adalah *DATAEXPORT* . Untuk dataexport 1 sampai 4 merupakan data total mesin1 dan dataexport a sampai e data total untuk mesin 2. Dan untuk data failure dapat dilihat pada Main code 3.4 Sistem Monitoring HMI, yaitu dengan memakai instruksi *call fdataexport*.

<pre> Private Sub DATAEXPORT() SPREADSHEET1.Cells(4, 4) = LTMONDAY.Caption End Sub Private Sub DATAEXPORT1() SPREADSHEET1.Cells(4, 5) = LTTUESDAY.Caption End Sub Private Sub DATAEXPORT2() SPREADSHEET1.Cells(4, 6) = LTWEDNESDAY.Caption End Sub Private Sub DATAEXPORT3() SPREADSHEET1.Cells(4, 7) = LTTTHURSDAY.Caption End Sub Private Sub DATAEXPORT4() SPREADSHEET1.Cells(4, 8) = LTFRIDAY.Caption End Sub </pre>	<pre> Private Sub DATAEXPORTA() SPREADSHEET1.Cells(9, 4) = MTMONDAY.Caption End Sub Private Sub DATAEXPORTB() SPREADSHEET1.Cells(9, 5) = MTTUESDAY.Caption End Sub Private Sub DATAEXPORTC() SPREADSHEET1.Cells(9, 6) = MTWEDNESDAY.Caption End Sub Private Sub DATAEXPORTD() SPREADSHEET1.Cells(9, 7) = MTTTHURSDAY.Caption End Sub Private Sub DATAEXPORTE() SPREADSHEET1.Cells(9, 8) = MTFRIDAY.Caption End Sub </pre>
--	---

Gambar 3.36 Pemrograman VB6 Main 6

Konsep dari pemrograman ini adalah dengan menyamakan nilai dari label total dengan *cells* pada MS.Excel 2003. nilai dari *cells* pada listing program diatas adalah nilai dari row dan column (*cells (row,column)*). Ketika nilai total mesin 1 pada monitoring HMI menunjukkan nilai 10 pada hari senin maka nilai tersebut juga akan ditulis pada baris 4, kolom 4 pada spreadsheet 1 di MS.excel. Jika nilai total mesin 2 pada monitoring HMI menunjukkan nilai 20 pada hari selasa maka nilai tersebut juga akan ditulis pada deret 9, kolom 5 pada spreadsheet 1 di MS.excel.

Proses kerja Last Delay (LD) pada tampilan monitoring HMI sesuai dengan List code 3.7 main 5, bahwa semua TimerLD akan aktif saat nilai Time Process berada pada posisi 0 yaitu pada saat proses etsa PCB berada pada tahap 4 dan pada saat tahap 5 yang berdurasi 10 detik maka proses last delay bekerja. Pada baris kedua list code main 5 (*LTPMonday.caption = 0*) merupakan instruksi yang membuat nilai Time Process bernilai nol. Begitu pula untuk hari Selasa, Rabu, Kamis, dan Jumaat.

Pada Main Code 3.1 Sistem Monitoring HMI yang merupakan keseluruhan dari pemrograman pada visual basic 6 ini terdapat bagian *Date & Time* yang menampilkan tanggal dan waktu pada kondisi yang sebenarnya, dan bagian *Introduction Machine*.

Namun kedua bagian ini merupakan bagian yang tidak terhubung dengan sistem Interfacing, yang mana bagian ini merupakan bagian tambahan yang digunakan untuk memudahkan operator produksi dalam bekerja.

Bagian Date & Time pada monitoring HMI terdapat pada Main Code 3.7 Sistem Monitoring HMI bagian 7 (Page 14). Pada bagian sub program berdiri sendiri sehingga tidak terhubung dengan sistem pemrograman interfacing lainnya. Subprogram ini berfungsi untuk menampilkan waktu, hari, tanggal, dan tahun yang sebenarnya.

Sedangkan pada bagian Introduction Machine terdapat pada Main Code 3.9 Sistem Monitoring HMI bagian 9. Prinsip kerja dari sistem source code ini adalah dengan menggabungkan beberapa gambar mesin yang sama, tetapi mempunyai sistem kerja pada gambar yang berbeda. Sehingga Gambar pada simulasi bagian ini dapat terlihat bergerak. Hal ini dilakukan dengan cara menghidupkan dan mematikan gambar pada simulasi yang diatur melalui timer 15 sampai timer 24 yang sudah ada pada fasilitas Visual basic 6.

3.5 Perhitungan Daya

Pada bab ini akan dibahas mengenai perkiraan nilai daya yang dihasilkan PLC untuk mengatur kondisi satu mesin dengan daur aktif 50%, 66.67%, dan 83.33%. Output dari PLC dihasilkan dari Power Supply 24 Vdc dengan asumsi arus (I) 0,04 Ampere. Berikut ini merupakan perkiraan dari perhitungan daya dari beberapa kondisi dalam 1 mesin.

1. Kondisi mesin dengan daur aktif 50% (C0)

Nilai dari Ton adalah 0,3s

Nilai dari Toff adalah 0,3s

Maka perkiraan daya outputnya (Pout) adalah:

$$V_{out} = \frac{T_{on}}{T_{on} + T_{off}} \times V_{in}$$

$$V_{out} = \frac{0,3}{0,3 + 0,3} \times 24$$

$$V_{out} = 12 \text{ V dc}$$

$$P_{out} = 12 \text{ V dc} \times 0,04 \text{ A}$$

$$\mathbf{0,48\text{watt}}$$

64

2. Kondisi mesin dengan daur aktif 66,67% (C1)

Nilai dari Ton adalah 0,4s

Nilai dari Toff adalah 0,2s

Maka perkiraan daya outputnya (Pout) adalah:

$$V_{out} = \frac{T_{on}}{T_{on} + T_{off}} \times V_{in}$$

$$V_{out} = \frac{0,4}{0,4 + 0,2} \times 24$$

$$V_{out} = 16 \text{ V dc}$$

$$P_{out} = 16 \text{ V dc} \times 0,04 \text{ A}$$

0,64 watt

3. Kondisi mesin dengan daur aktif 83,33% (C2)

Nilai dari Ton adalah 0,5s

Nilai dari Toff adalah 0,1s

Maka perkiraan daya outputnya (Pout) adalah:

$$V_{out} = \frac{T_{on}}{T_{on} + T_{off}} \times V_{in}$$

$$V_{out} = \frac{0,5}{0,5 + 0,1} \times 24$$

$$V_{out} = 20 \text{ V dc}$$

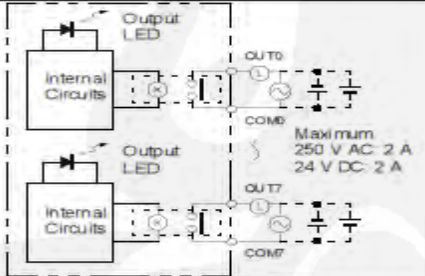
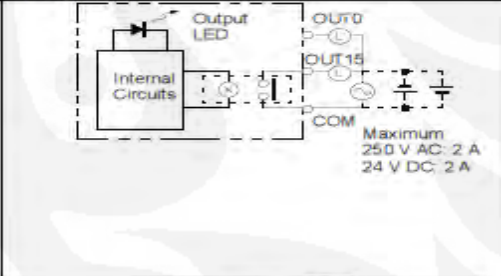
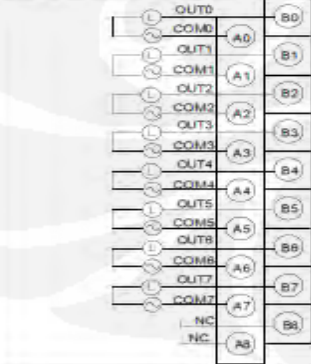
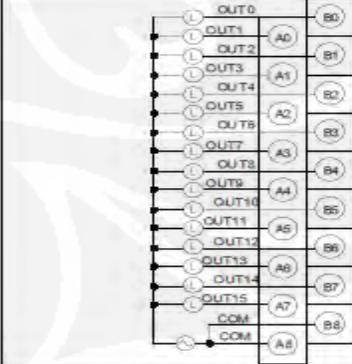
$$P_{out} = 20 \text{ V dc} \times 0,04 \text{ A}$$

0,8 watt

Dari nilai perhitungan daya diatas terbukti bahwa sistem yang dipakai dalam proses ini dapat menghemat pemakaian daya, sehingga sisa (*space*) yang sangat berguna untuk penggunaan beban-beban yang lainnya, seperti penambahan sinyal-sinyal mikrokontroller kedepannya ataupun penambahan sistem-sistem lainnya yang terhubung dengan bagian terpenting dari sistem keseluruhan ini yaitu PLC.

3.6 Sistem Pendukung Lainnya

Modul output PLC CQM 1 adalah OC222 yang mempunyai sistem unit kontak. Dengan spesifikasi seperti pada tabel 3.7 dibawah ini:

Item	CQM1-OC221	CQM1-OC222
Name	8-point Contact Output Unit	16-point Contact Output Unit
Max. Switching Capacity	2 A, 250 V AC ($\cos\phi = 1$) 2 A, 250 V AC ($\cos\phi = 0.4$) 2 A, 24 V DC (16 A/Unit)	2 A, 250 V AC ($\cos\phi = 1$) 2 A, 250 V AC ($\cos\phi = 0.4$) 2 A, 24 V DC (8 A/Unit)
Min. Switching Capacity	10 mA, 5 V DC	10 mA, 5 V DC
Relay	G6D-1A	G6D-1A
Service Life of Relay	Electrical: 300,000 operations (resistive load) 100,000 operations (inductive load) Mechanical: 20,000,000 operations (See note.)	Electrical: 300,000 operations (resistive load) 100,000 operations (inductive load) Mechanical: 20,000,000 operations (See note.)
ON Delay	10 ms max.	10 ms max.
OFF Delay	5 ms max.	5 ms max.
No. of Outputs	8 points (independent commons)	16 points (16 points/common, 1 circuit))
Internal Current Consumption	430 mA max. at 5 V DC	850 mA max. at 5 V DC
Weight	200 grams max.	230 grams max.
Circuit Configuration		
Terminal Connections		

Tabel 3.7 Spesifikasi Modul Output PLC CQM1

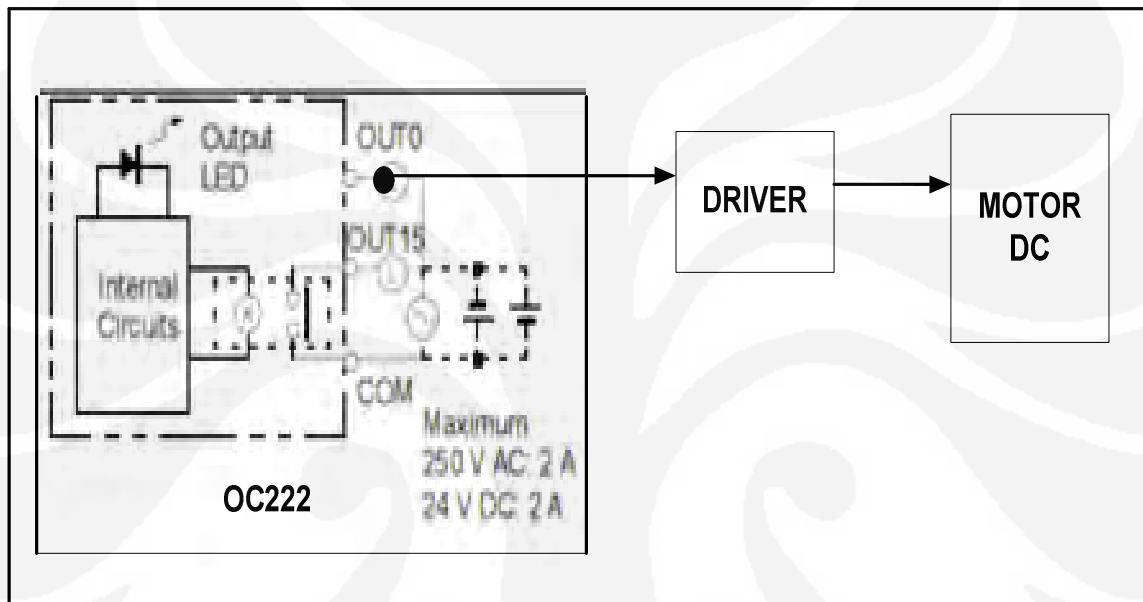
Sesuai dengan rangkaian diatas maka motor DC pada mesin 1 terhubung ke 100.03 out3 dan mesin 2 terhubung pada 100.08 out8. Modul output OC222 mempunyai batas maksimum *switching* 2A untuk 24Vdc dan batas minimum *switching* 10 mA, 5V dc. Sedangkan pada motor dc *geared* yang terhubung dengan mesin membutuhkan daya sekitar 0,73 watt dan tegangan 24 Vdc seperti pada tabel 3.8. Jadi sesuai dengan batas maksimum 2A, yaitu nilai arus yang dibutuhkan motor adalah 0,73 watt dibagi 24 Vdc, yaitu sekitar 0,03 A.

Tabel 3.8 merupakan spesifikasi dari motor DC Geared dengan berat sekitar 180 g jenis *low noise long life*. Untuk datasheet lengkapnya ada pada Lampiran W.

ITEM	GEARED MOTOR
NAME	TG-382430000-30K
Rated Volt (V)	24
No Load	
Current (mA)	≤18
Speed (r/min)	100
Load Torque	
Current (mA)	≤85
Speed (r/min)	83
Torque	
Kgf.cm	0,08
N.m	0,09
At maximum Efficiency	
Kgf.cm	3
N.m	0,29
Output Power (W)	0,73
Number Of Geared Trains	3
Gear Box Length (mm)	29

Tabel 3.8 Spesifikasi Output Motor DC

Selain itu pwm yang dikerjakan oleh PLC pada mesin ini bekerja pada perioda siklus besar dan frekuensi rendah. Perioda yang dikerjakan pada mesin ini adalah 0,6s dengan frekuensi sekitar 1,66 HZ, namun untuk perioda yang rendah dan frekuensi tinggi, dibutuhkan *driver* atau inverter sebelum masuk ke motor seperti pada gambar 3.42 dibawah. Pada umumnya untuk frekuensi tinggi pada motor DC digunakan driver *Smart Pheriperal Controller* (SPC-06). Namun untuk perioda sekitae 0,6 s keatas, PLC mampu untuk menghasilkan perbandingan Ton dan Toff dengan nilai tersebut.



Gambar 3.37 Blok Diagram Sistem Output PLC Dengan Frekuensi Tinggi

Driver pada blok diagram diatas akan diprgunakan untuk sistem PLC dengan pwm yang memiliki perioda kecil (ms) dan frekuensi tinggi (KHZ). Sedangkan untuk mesin Etching PCB tidak membutuhkan frekuensi yang besar dan perioda kecil, karen PLC masih mampu menghasilkan PWN dengan perioda 0,6s keatas.

BAB IV

PENGUJIAN DAN ANALISA SISTEM

Pada bab 4 ini akan dibahas hasil dari perancangan alat dimana bertujuan untuk mengetahui sejauh mana kerja alat tersebut. Untuk pengambilan data dilakukan dalam beberapa tahap, yaitu pengamatan nilai angka yang masuk pada PC dari Mikrokontroler yang dikirim dari PLC. Yang bertujuan untuk mengetahui bahwa angka dari port berapa saja yang akan diolah oleh VB6, sebagai inti dari sistem antarmuka. Tahap kedua adalah pengamatan *time process* tampilan HMI yang akan dibandingkan dengan nilai *time process actual* pada *stop watch* pada jam digital yang bertujuan untuk mengetahui keakuratan sistem *Interfacing* HMI. Dan tahap ketiga adalah pengambilan data banyaknya jumlah goresan pada PCB saat mesin bekerja.

Pengamatan nilai angka sebagai informasi ke PC dilakukan dengan memasukkan input keempat P0 mikrokontroler dengan bergantian, seperti pada tabel 4.1. perubahan dari masukkan ke mikrokontroler mengakibatkan tampilan pada Hyper terminal PC juga berubah.

4.1 Pengujian Sinyal *Bridge*

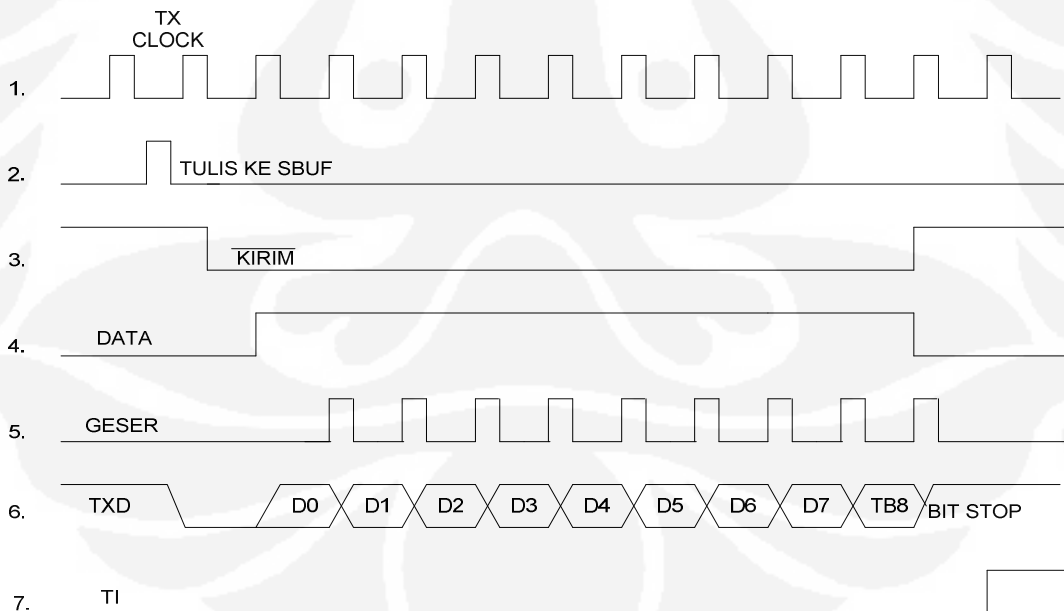
Pengamatan data yang masuk ke PC dapat dilihat pada fasilitas dari windows Xp yaitu Hyper Terminal. Data Bridge yang masuk ke PC terdiri dari dua *Gate*, yaitu com1 untuk mesin2 dan com3 untuk mesin 1. Hasil dari pengamatan dapat dilihat pada tabel 4.1.

NO	PORT PLC	BIT P0 uC	CONDITION	TAMPILAN HYPER TERMINAL (COM1)	TAMPILAN HYPER TERMINAL (COM3)
1	000.03/000.05	#11111110b	uC1(C0)		0
2	TIMER	#11111101b	uC1(C1)		1
3	TIMER	#11111011b	uC1(C2)		2
4	000.00	#10111111b	uC1(F)		6
5	000.04/000.06	#11111110b	uC2(C0)	0	
6	TIMER	#11111101b	uC2(C1)	1	
7	TIMER	#11111011b	uC2(C2)	2	
8	000.01	#10111111b	uC2(F)	6	

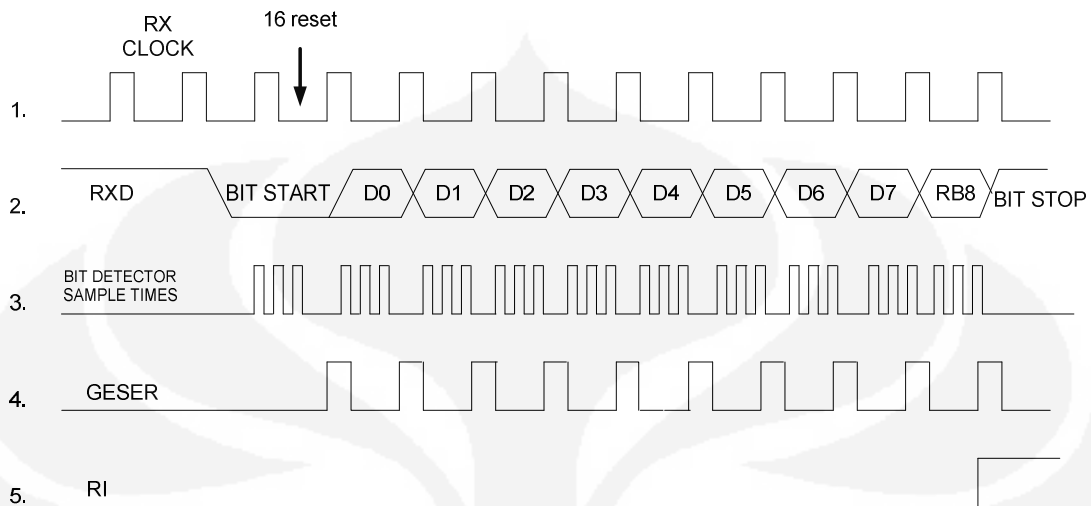
Tabel 4.1. Data Nilai Output *Bridge*

Nilai-nilai yang terlihat pada tabel 4.1 merupakan nilai yang nantinya diolah oleh VB6 yang selanjutnya dapat ditampilkan pada sistem monitoring HMI. Pada kolom BIT PO mikrokontroller, nilai bit #11111110b mempunyai arti angka terakhir merupakan LSB (*Least Significant Bit*) atau P0.0 *active low* sedangkan angka pertama merupakan MSB (*Most Significant Bit*) atau P0.7 *active low*. Contoh tampilan lengkap Hyper Terminal dapat tergambar pada Gambar IV.1.1 dan IV.1.2. Sebelas bit yang dikirim (melalui TXD) atau diterima (melalui RXD) dalam susunan sebuah bit start (=0), 8 bit data (LSB urutan pertama), bit data ke-9 yang terprogram dan sebuah bit stop (=1). Sebelum pengiriman, bit data ke-9 (TB8) dapat diisi 0 atau 1. Pada saat penerimaan, bit stop masuk ke dalam ke dalam RB8 di register SCON. *Baud rate* dapat dipilih 1/32 atau 1/64 dari frekuensi kristal untuk mode 2, sedangkan mode 3 dapat memiliki baud rate yang bervariasi yang dihasilkan baik dari timer 1 maupun 2, bergantung dari kondisi TCLK dan RCLK. Seperti yang tergambar pada diagram waktu IV.1.1 dan IV.1.2 pada **Mikrokontroller AT89s51.**

Pengiriman atau transmisi data diinisialisasi dengan instruksi yang melibatkan SBUF sebagai register tujuan. Sinyal 2 akan mengirim TB8 ke posisi 9 dari register geser pengiriman dan memberikan tanda untuk TX bahwa ada permintaan pengiriman sinyal.



Gambar 4.1 Diagram Waktu Serial Mode 2 (Kirim)



Gambar 4.2 Diagram Waktu Serial Mode 2 (Terima)

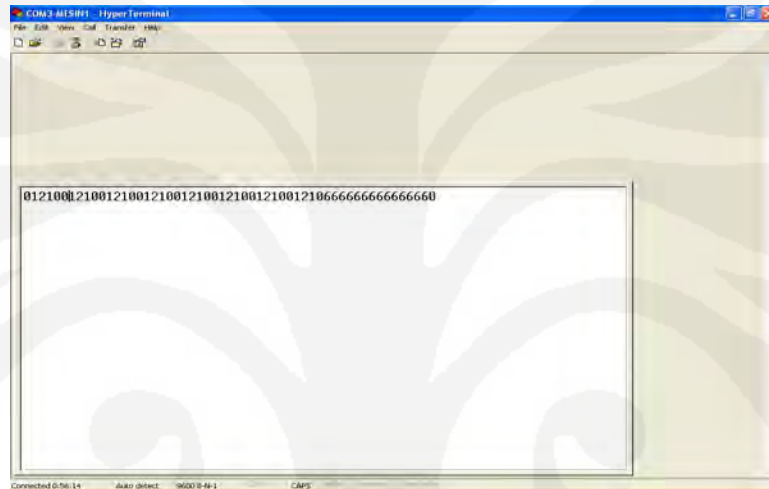
Transmisi data dimulai pada saat sinyal KIRIM *active low* diaktifkan (=0) yang ditunjukkan pada diagram waktu IV.1.1 bagian ketiga yang kemudian akan menempatkan bit start pada jalur TXD. Satu bit kemudian, DATA diaktifkan sekaligus akan mengaktifkan bit keluaran dari register geser pengiriman TXD. Pulsa penggeseran yang pertama akan muncul 1 bit kemudian. Pulsa geser yang pertama akan mengisi 1 (bit stop) ke posisi 9 dari register geser, setelah itu hanya data 0 yang diisikan. Dengan demikian, saat bit data digeser keluar kanan, data 0 diisi ke kiri. Saat TB8 pada posisi keluaran dari register geser, maka bit stop berada dikiri MSB dan sisa posisi disebelah kirinya berisi 0. Kondisi ini memberikan tanda kepada unit KONTROL TX untuk melakukan penggeseran yang terakhir kemudian menon-aktifkan KIRIM dan menset TI. Hal ini terjadi pada saat siklus mesin kesebelas setelah sinyal 2 diagram waktu IV.1.

Penerimaan dimulai saat terjadi transisi 1 ke 0 pada RXD. Dalam hal ini, RXD dicuplik dengan laju 16 kali baud rate yang ditetapkan. Saat transmisi deteksi, pencacah pembagi 16 direset dan 1fff dituliskan kedalam register masukkan.

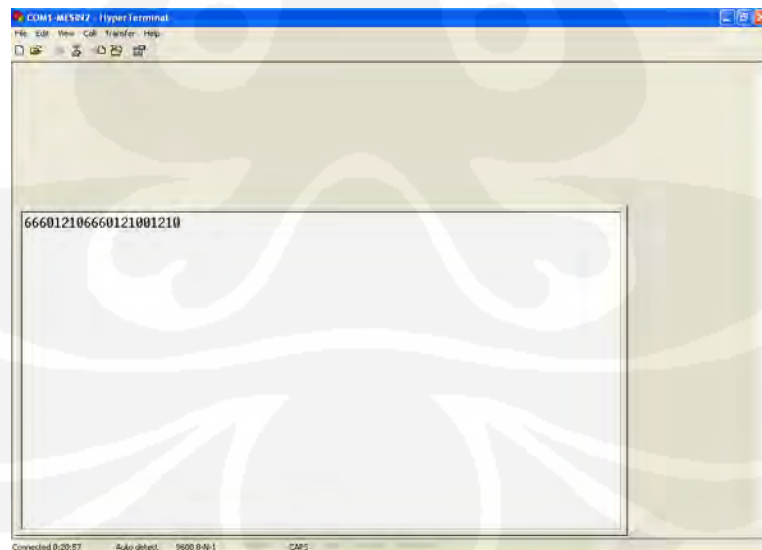
Saat data masuk dari kanan, data "1" digeser keluar ke kiri. Pada saat bit start tiba di posisi paling kiri dalam register geser (sebuah register 9-bit dalam mode 2), maka akan memberi tanda pada KONTROL RX untuk melakukan penggeseran terakhir, isi SBUF

dan RB8 serta menseset RI, sinyal-sinyal ini dihasilkan, jika dan hanya jika kondisi berikut ini bertepatan dengan waktu pulsa geser terakhir. Yaitu saat RI=0 dan SM2=0 atau bit ke-9 yang diterima adalah 1.

Apabila salah satu dari kedua kondisi tersebut tidak terpenuhi, maka *frame* data yang diterima akan hilang (*frame error*). Namun jika kedua kondisi tersebut dipenuhi, maka bit stop akan ditampung ke RB8, data 8-bit masuk ke SBUF dan RI diaktifkan. Setelah itu unit akan melihat atau menunggu transisi 1 ke 0 lagi pada jalur RXD (data berikut).

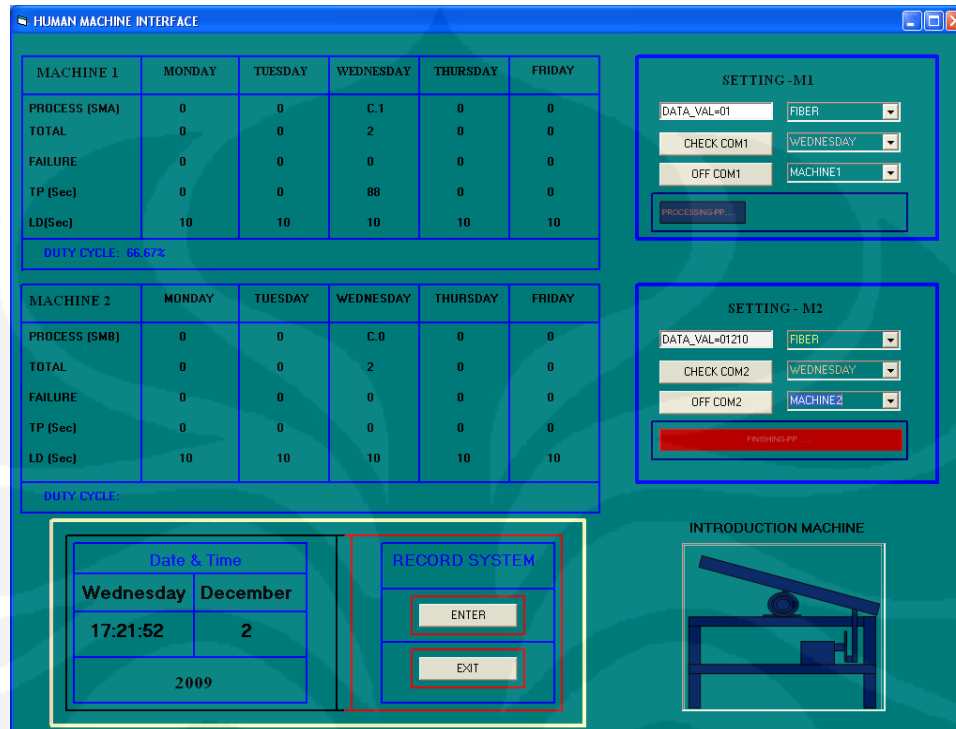


Gambar 4.3 Tampilan Data Masuk Ke COM3



Gambar 4.4 Tampilan Data Masuk Ke COM1

4.2 Pengujian *Time Process*



Gambar 4.5 Tampilan *Time Process*

Pengujian dilakukan dengan membandingkan nilai TP pada sistem HMI dan *stop watch* jam digital Casio, yang didalam tabel 4.2 terdapat di kolom TP *Actual*. Nilai dari TP ini adalah nilai dari 10 produk PCB dengan 5 jenis rangkaian yang berbeda. Dan nilai-nilai tersebut akan ditampilkan pada tabel 4.2.

Data pada tabel 4.2 diambil pada hari Rabu 2 Desember 2009 pada jam 16.00. Adapun rangkaian yang diambil ada 5 jenis yang hampir mewakili dari keseluruhan rangkaian pada umumnya. Yaitu pada ukuran 19 X 9cm kebawah, dan jenis PCB yang diambil adalah *Fiber* dan *PCP (Phenolic Cotton Paper)*. Kolom % *Failure* merupakan nilai kesalahan yang akan diperkirakan saat PCB selesai diproses. Jalur yang putus atau tidak berhasil menempel pada PCB akan dianggap sebagai failure dari mesin. Pada gambar IV.2 merupakan tampilan HMI saat pengambilan data PCB fiber yang kelima.

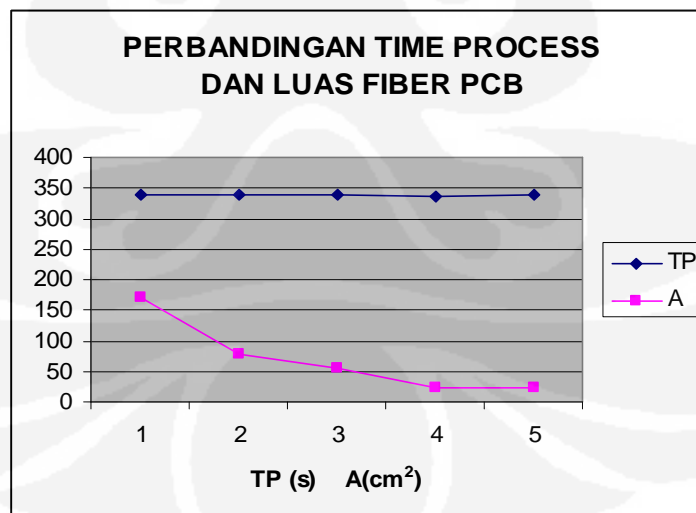
Untuk PCB *fiber* 1,5, dan 9 dikerjakan di mesin yang pertama, sedangkan nomor 3 dan 7 dikerjakan di mesin yang kedua, demikian pula pada PCB jenis PCP.

NO	SERIES	P X I (CM)	PCB	TIME PROCESS (S) ACTUAL	TIME PROCESS (S) HMI DISPLAY	%FAILURE	DATE	TIME
1	DRIVER MOTOR DC	19 X 9	FIBER	340	340	0	12/2/2009	16.00
2	DRIVER MOTOR DC	19 X 9	PCP	300	300	0	12/2/2009	16.00
3	POWER SUPPLY	13 X 6	FIBER	338	338	0	12/2/2009	16.15
4	POWER SUPPLY	13 X 6	PCP	300	300	0	12/2/2009	16.15
5	LCD DISPLAY	9 X 6	FIBER	338	338	0	12/2/2009	16.30
6	LCD DISPLAY	9 X 6	PCP	300	300	0	12/2/2009	16.30
7	DRIVER M2	6 X 4	FIBER	337	337	0	12/2/2009	16.45
8	DRIVER M2	6 X 4	PCP	298	298	0	12/2/2009	16.45
9	BRIDGE PWM AC	5.9 X 4	FIBER	338	338	0	12/2/2009	17.00
10	BRIDGE PWM AC	5.9 X 4	PCP	298	298	0	12/2/2009	17.00

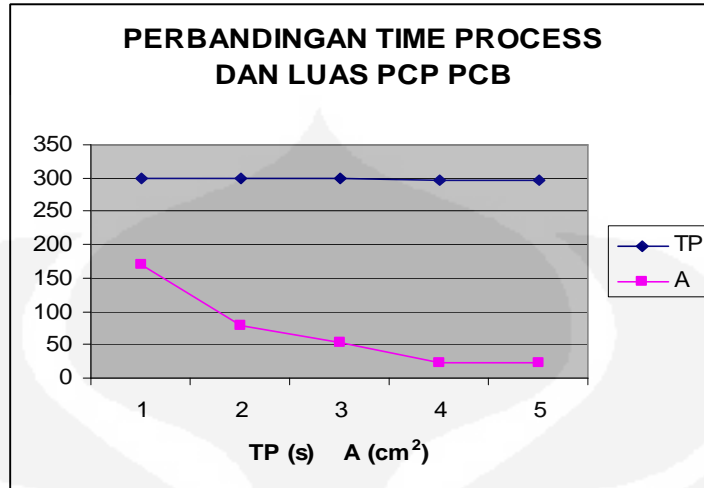
Tabel 4.2 Data *Time Process*

Dari tabel diatas dapat diketahui bahwa nilai TP Actual sama dengan nilai TP pada tampilan HMI sistem. Sehingga data yang diambil pada keadaan sebenarnya, sesuai dengan data pada tampilan HMI.

Sistem TP pada tampilan HMI menggunakan teknik dengan cara menjalankan fasilitas timer yang tersedia pada VB6 pada saat proses awal mesin aktif atau pada kondisi C0. dengan nilai timer 1 detik dan kenaikan 1 angka, atau pada program dapat dilihat dengan instruksi $val(LTP\ MONDAY.caption + 1)$. Maka sudah dapat dipastikan hasil dari perhitungan TP pada tampilan HMI adalah *valid*.



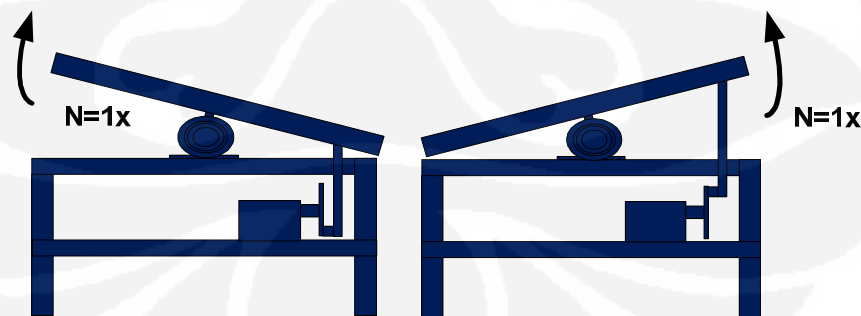
Gambar 4.6 Grafik Perbandingan TP Dan A (FIBER)



Gambar 4.7 Grafik Perbandingan TP Dan A (PCP)

Sedangkan perubahan dari nilai TP pada setiap perbedaan rangkaian tidak terlalu *significant*. Seperti yang sudah ditampilkan pada grafik perbandingan 4.6 dan 4.7, bahwa perubahan garis A atau luas PCB sama sekali tidak membuat perubahan tinggi dari garis biru atau TP. Bentuk dari Garis biru pada grafik itu hampir mendekati lurus. Dapat dipastikan untuk semua ukuran PCB yang dibawah 25 X 10 cm nilai TP pada tampilan HMI adalah mendekati. Sehingga diketahui bahwa nilai dari *Time Process* tidak tergantung pada ukuran luas PCB.

TP tergantung dari banyaknya perkiraan jumlah goresan larutan pada PCB (N) tiap kondisi, mulai dari C0,C1,C2,C2, sampai C0 kembali (pada gambar 4.8), Jumlah goresan dapat diperkirakan melalui gerakan dudukan mesin. Pada mesin yang sudah terprogram dengan sistem ini memiliki jumlah goresan yang ada pada tabel 4.3.



Gambar 4.8 Perkiraan Jumlah Goresan PCB

NO	CONDITION	N	WAKTU CONSTANT HMI DISPLAY	DATE
1	M1(C0)	96x	72s	12/2/2009
2	M1(C1)	126x	72s	12/2/2009
3	M1(C2)	130x	72s	12/2/2009
4	M1(C0)	96x	72s	12/2/2009
5	M1(C1)	126x	72s	12/2/2009
6	M1(C2)	130x	72s	12/2/2009

Tabel 4.3 Perkiraan Jumlah Goresan PCB

Dari tabel diatas dapat diketahui bahwa jumlah goresan tergantung dari perbedaan kondisi-kondisi mesin yang ditandai dari CO,C1, dan C2. Perubahan ketiga kondisi tersebut merupakan peribahan *duty cycle* dari motor penggerak mesin, semakin tinggi *duty cycle* motor maka rpm motor atau putaran motor setiap menitnya semakin tinggi dan mengakibatkan jumlah goresan PCB(N) semakin banyak.

BAB V

KESIMPULAN

Dari hasil pembahasan pada bab-bab sebelumnya didapat beberapa kesimpulan yaitu:

1. PLC dapat menggabungkan dan mengontrol lebih dari 1 mesin yang mempunyai aktuator sederhana (minimal) seperti mesin Etching PCB, namun untuk mesin yang mempunyai aktuator *complex* seperti mesin Las otomatis (*Schlater*), Pemotong besi (*wafios R43b*), PLC harus mengontrol hanya 1 mesin saja.
2. Sistem antarmuka PLC ke Mikrokontroller dilakukan dengan menghubungkan relay-relay, hal ini dikarenakan adanya perbedaan tegangan antara output PLC yaitu 24Vdc dan input mikrokontroller sekitar 5 sampai 12Vdc.
3. Sistem antarmuka PLC ke PC dapat dilakukan melalui *converter* PLC CPM1a beserta software yang terhubung langsung dengan CPM1a, namun teknik seperti ini membutuhkan biaya (*cost*) yang besar. Dengan memakai mikrokontroller AT89S51 sebagai pengganti *converter* CPM1a maka tidak membutuhkan biaya yang besar untuk membuat antarmuka dari PLC ke PC.
4. *Converter* PLC CPM1a hanya untuk PLC CQM1 CPU 4x, untuk PLC jenis lainnya menggunakan *converter* jenis lain. Sedangkan AT89S51 dapat dipakai di semua jenis PLC. Hanya dengan memakai port sisa dari input maupun output dari PLC yang akan dipakai, baik itu jenis Mitshubishi, Omron, Siemens, ABB, dan jenis lainnya yang dipakai di Industri.
5. Kelemahan dari sistem antarmuka PLC ke PC ini adalah AT89S51 hanya menyediakan satu jalur pengiriman dan penerimaan P3.0(RXD) dan P3.1(TXD) jadi untuk mengirim informasi dua mesin harus menggunakan dua mikrokontroller sekaligus, sehingga informasi yang diperoleh *valid*. Sehingga apabila dikemudian hari mesin ditambah menjadi tiga maka harus ada penambahan mikrokontroller atau memakai mikrokontroller jenis lain, yang menyediakan fasilitas jalur pengiriman dan penerimaan lebih banyak.

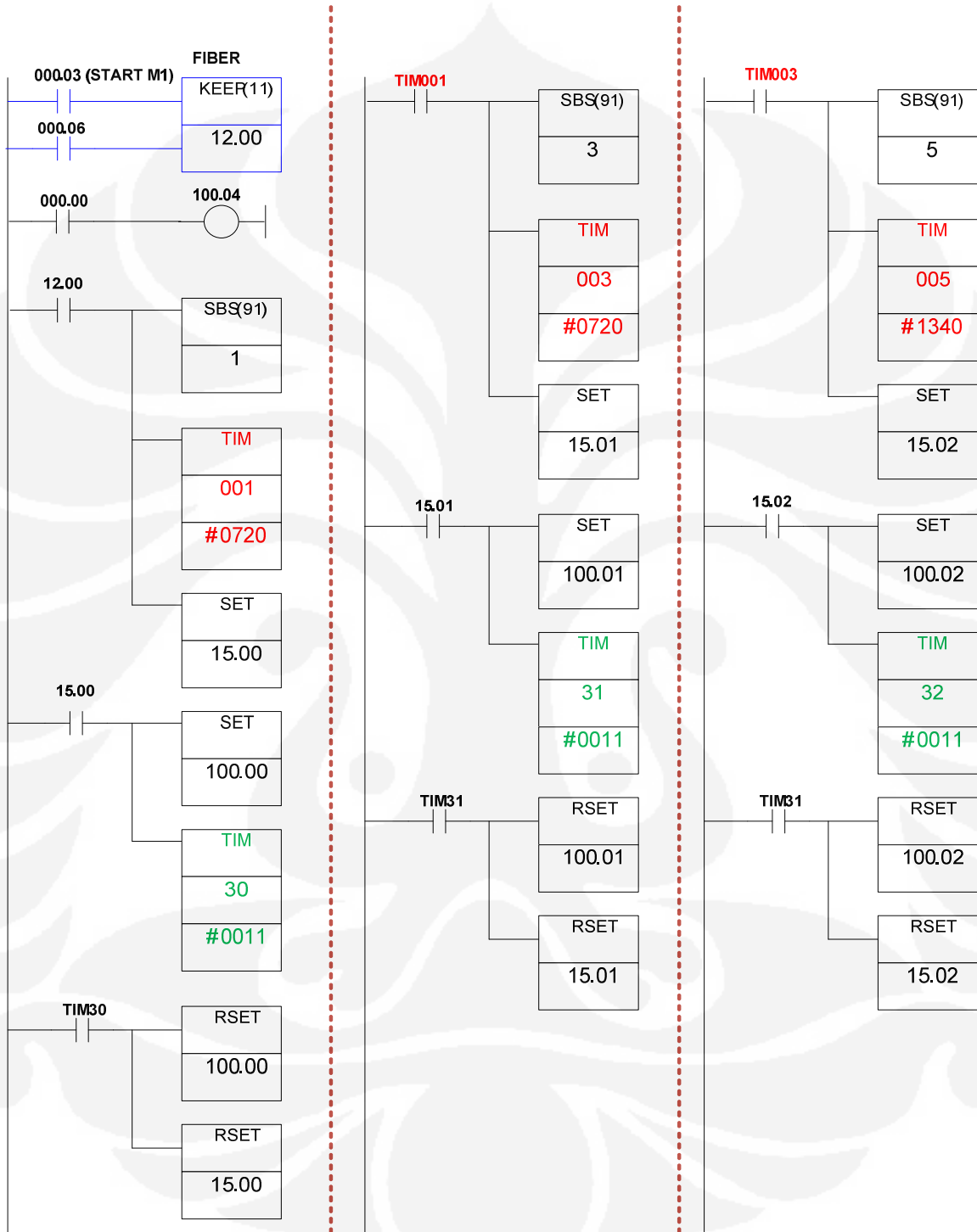
6. Tampilan HMI pada sistem ini dengan menggunakan VB6 yang membuka 2 commport sekaligus, yaitu mscomm1 untuk mesin 2 dan mscomm3 untuk mesin 1. Yang mana tampilan ini menunjukkan nilai total dan failure PCB dari hari Senin sampai Jumaat, sedangkan hari Sabtu dan Minggu dihitung lembur.
7. Sistem HMI ini dengan mengambil data yang dikirim dari mikrokontroler yang bernilai 0,1,2,dan 6 sebagai failure. Nilai 0,1,2 merupakan nilai yang akan diproses untuk untuk menampilkan nilai Total, Failure, TP, LD, di kolom Senin sampai Jumaat dengan cara membentuk subrutin-subrutin program atau dapat dikatakan *modular*.
8. Perubahan nilai luas pada PCB tidak berpengaruh terhadap perubahan nilai TP pada tampilan HMI. Namun untuk mempercepat proses pada pengerjaan PCB hanya dapat dilakukan dengan cara menambah jumlah goresan pada PCB yaitu dengan cara merubah sedikit mekanik mesin dengan meninggikan bagian dudukan ferric clorida dan memperbesar *Duty cycle*.
9. Sistem antarmuka ini dapat dikembangkan, sehingga data dari *plant* dapat diterima oleh bagian manajemen produksi. Dengan cara memasukkan sistem jaringan kedalam PC, yaitu menambah *.netframework* dan memakai Visual Basic 2005 yang sudah mempunyai fasilitas LAN.
10. Dengan membuat sistem online pada HMI, maka data dapat langsung diterima oleh bagian manajemen pengolahan data secara langsung.
11. Pada sistem antarmuka ini dapat dikembangkan menjadi lebih dari 2 mesin, dengan cara menambah *bridge* sistem atau mikrokontroler sebagai pengiriman sinyal. Namun harus sesuai dengan banyaknya port yang disediakan oleh PC/ laptop yang *compatible*.
12. Untuk pengembangan mesin lebih dari dua, dengan 1 sistem *Bridge* dapat dilakukan, namun kelemahannya adalah pengaktifan antara mesin 1 dan mesin lainnya tidak bisa dalam waktu yang bersamaan. Yaitu dengan cara menambah kecepatan baud rate (bps) tetapi tetap penyalaan mesin tidak boleh bersamaan, karena hanya 1 jalur pengiriman yang terpakai.

DAFTAR REFERENSI

1. Agfianto Eko Putra, *Pemrograman Mikrokontroller AT89S51*, Andi Offset Yogyakarta
2. Catur Edi Widodo, *Visual Basic 6.0*, Andi Offset Yogyakarta
3. Michael Halvorson, *Microsoft Visual Basic 6.0 Profesional Step by Step*, PT Elex Media Komputindo Jakarta
4. William Bolton, *Programmable Logic Controller*, ERLANGGA
5. PLC : http://en.wikipedia.org/wiki/Programmable_logic_controller, terakhir diakses 1 Desember 2009
6. HMI : <http://www.isa.org/HMI>, terakhir diakses 1 Desember 2009

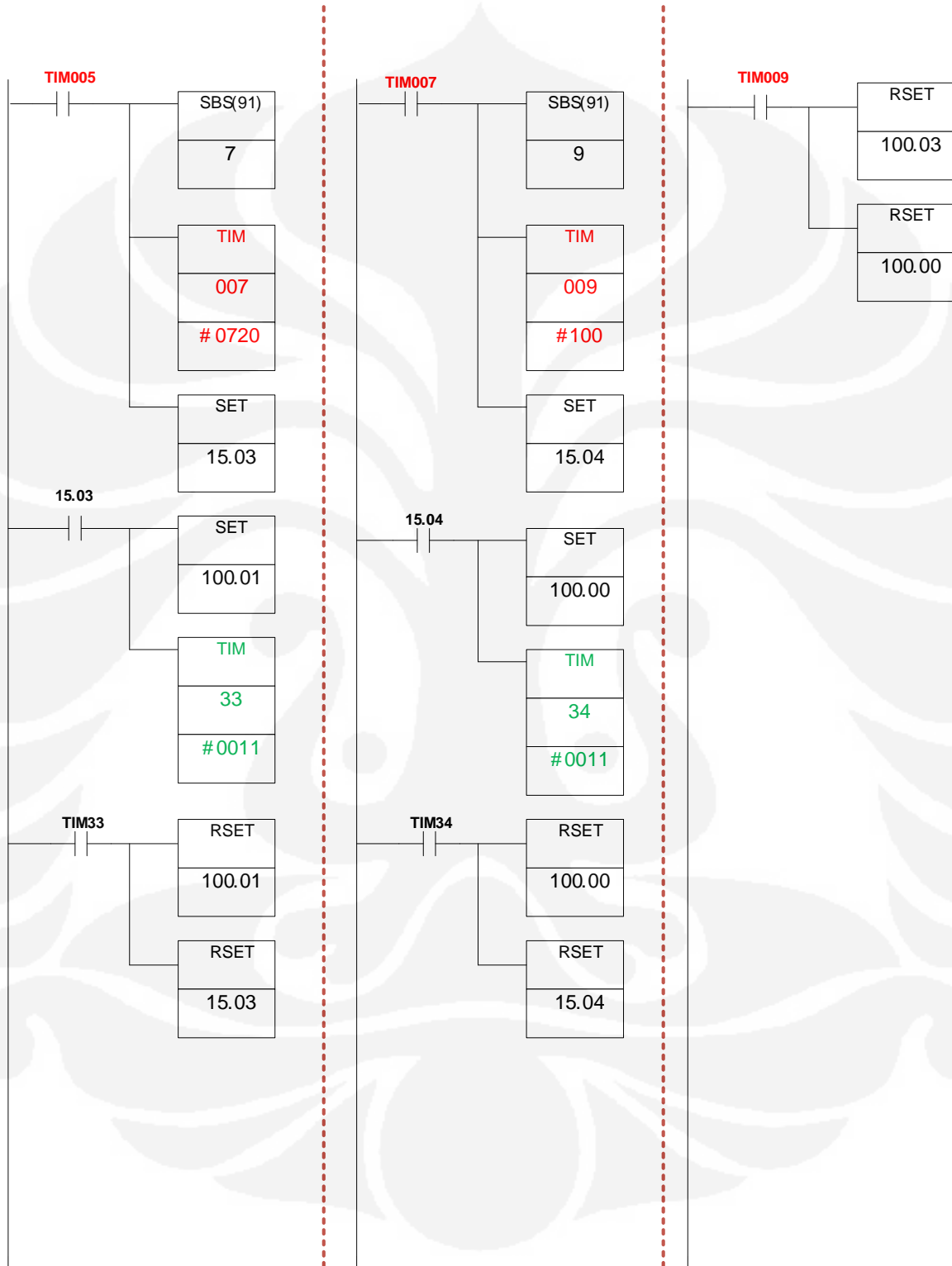
LAMPIRAN A

PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 1



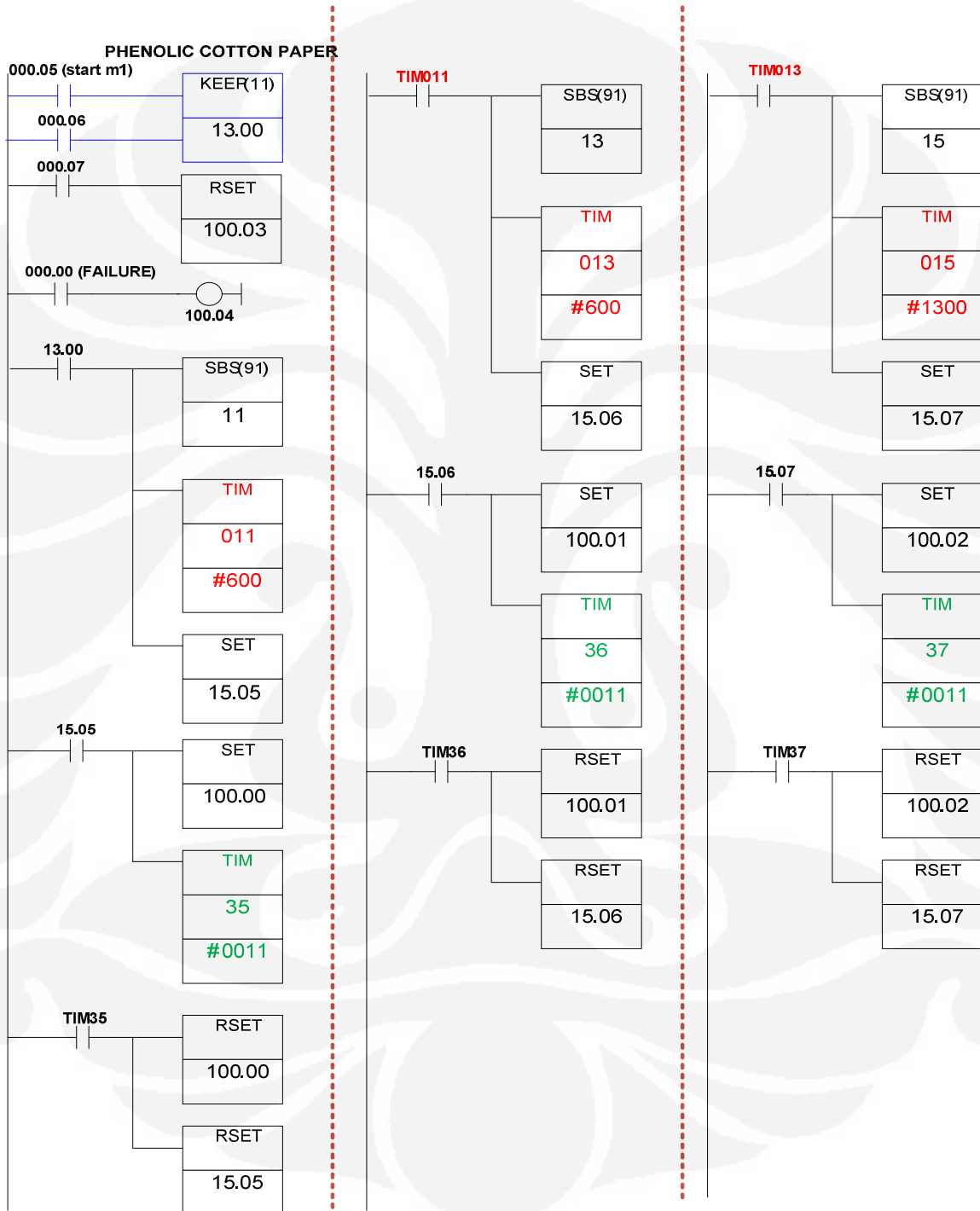
LAMPIRAN B

PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 2

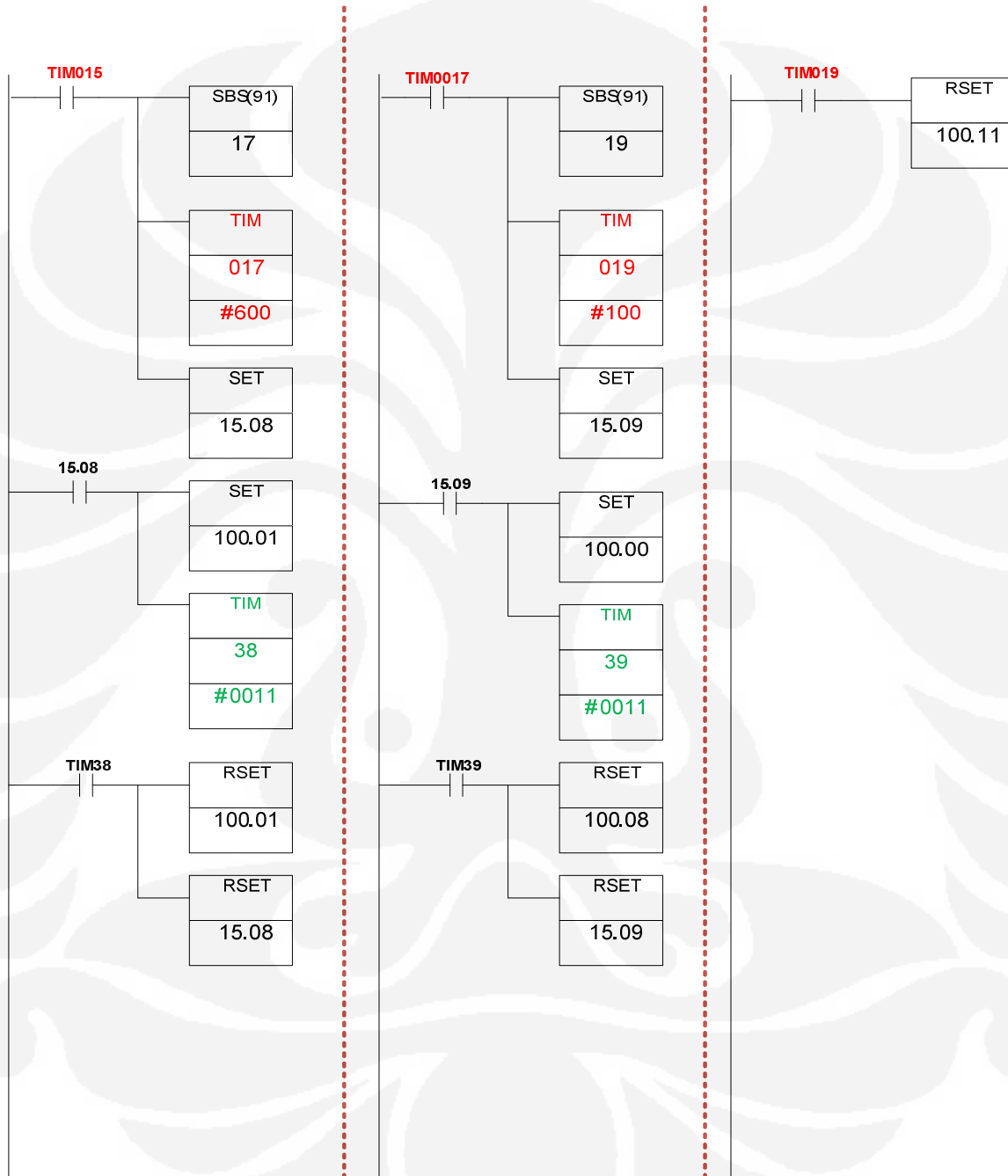


LAMPIRAN C

PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 3

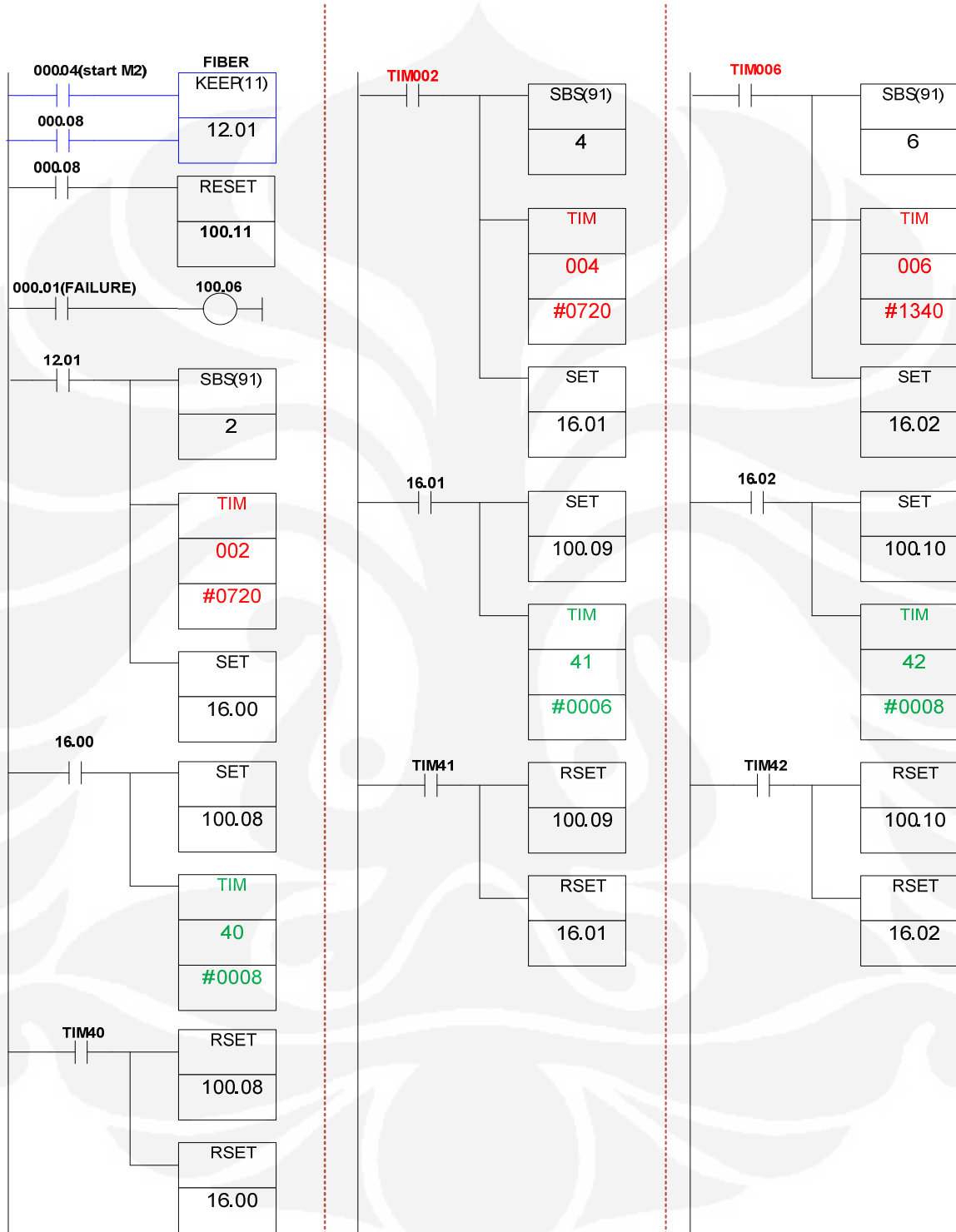


LAMPIRAN D
PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 4

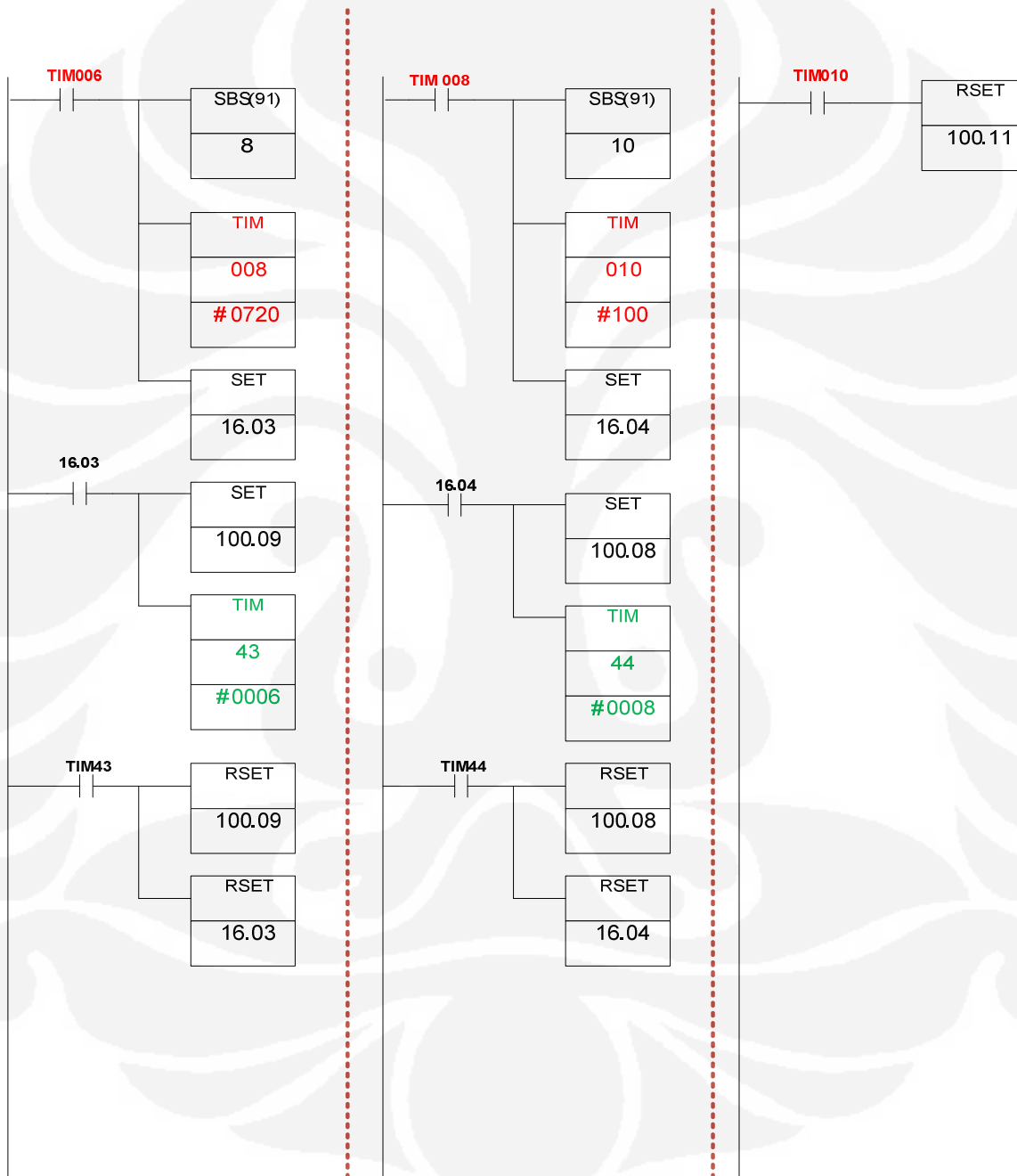


LAMPIRAN E

PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 5

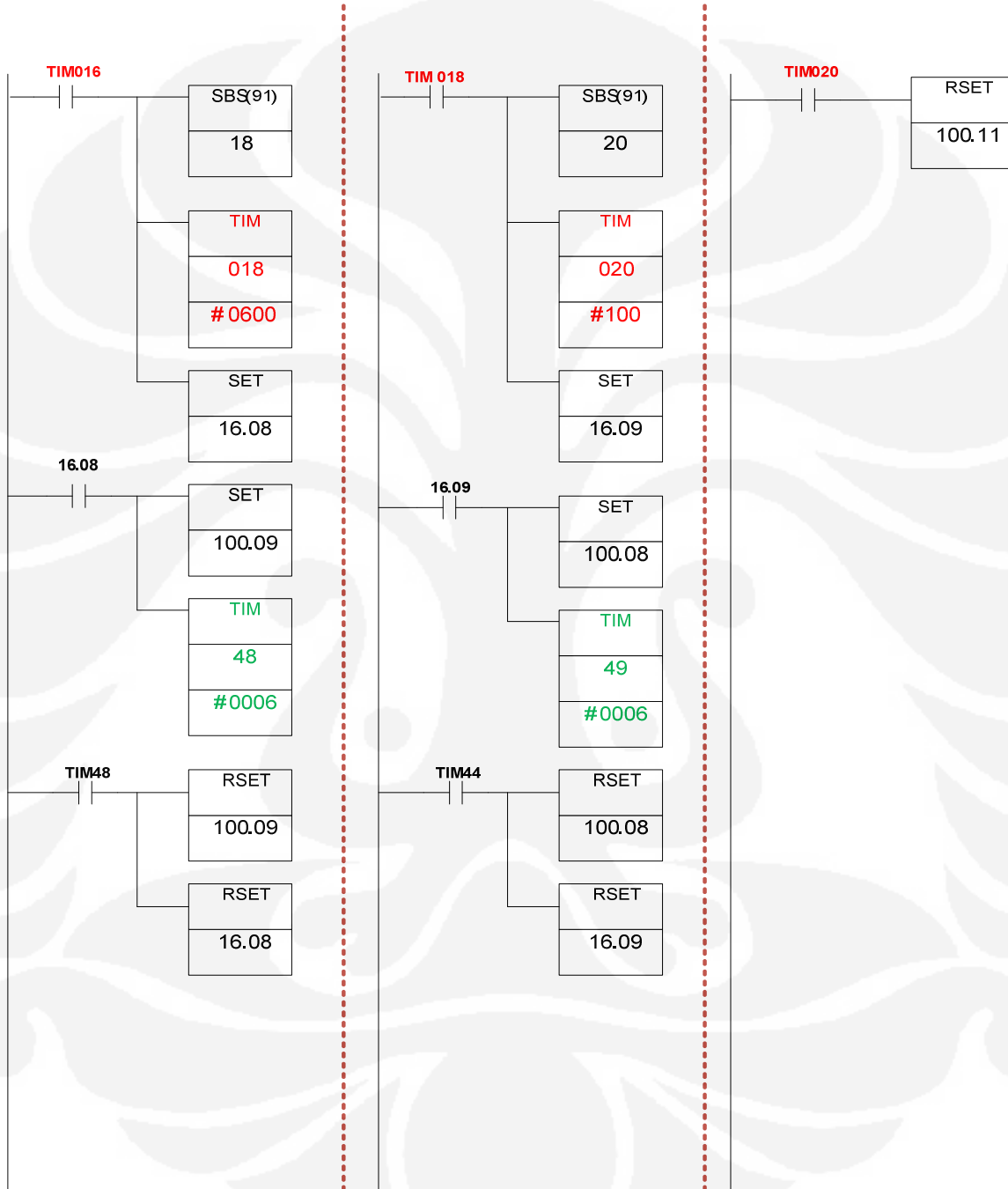


LAMPIRAN F
PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 6



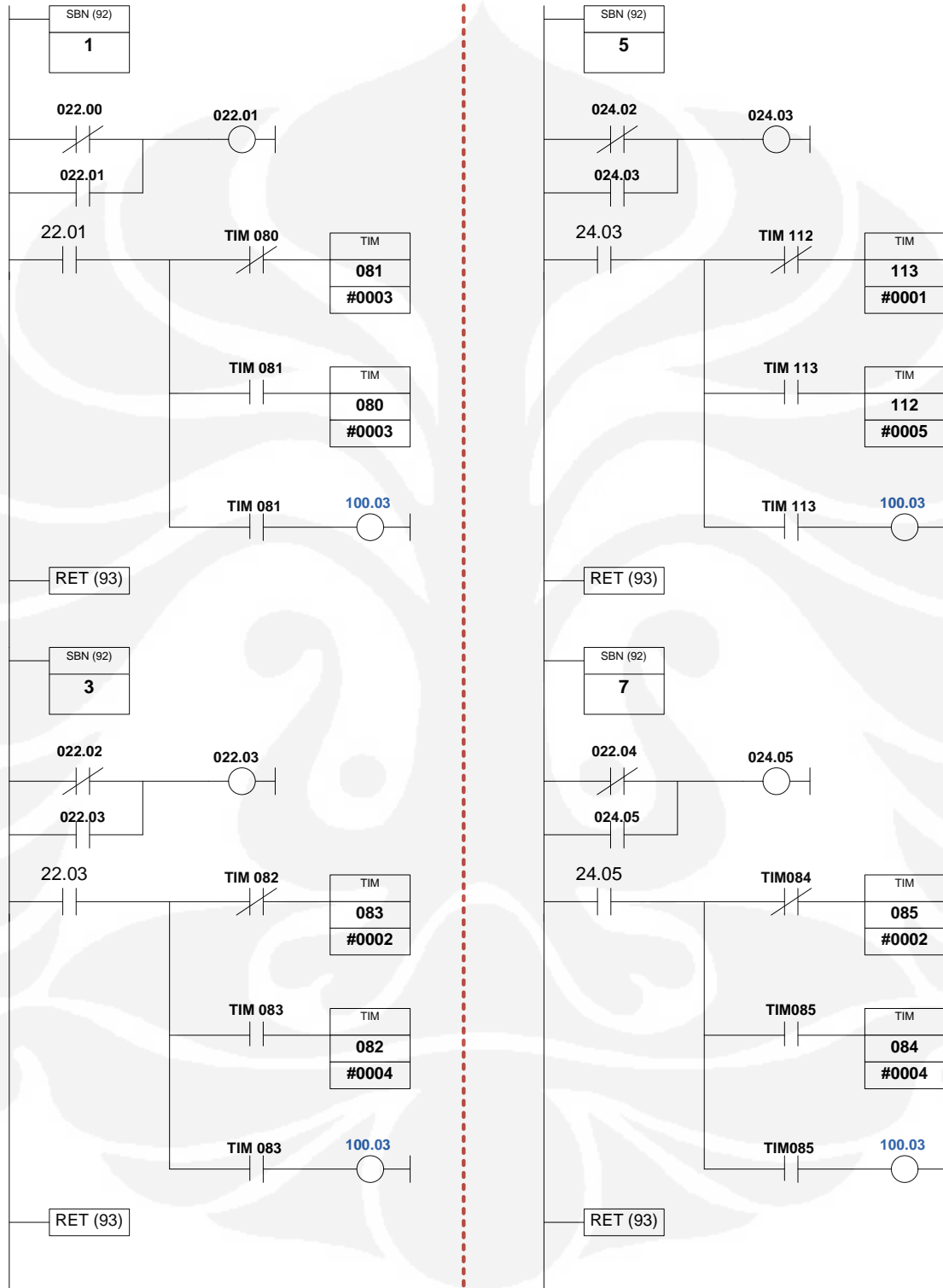
LAMPIRAN G

PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 8



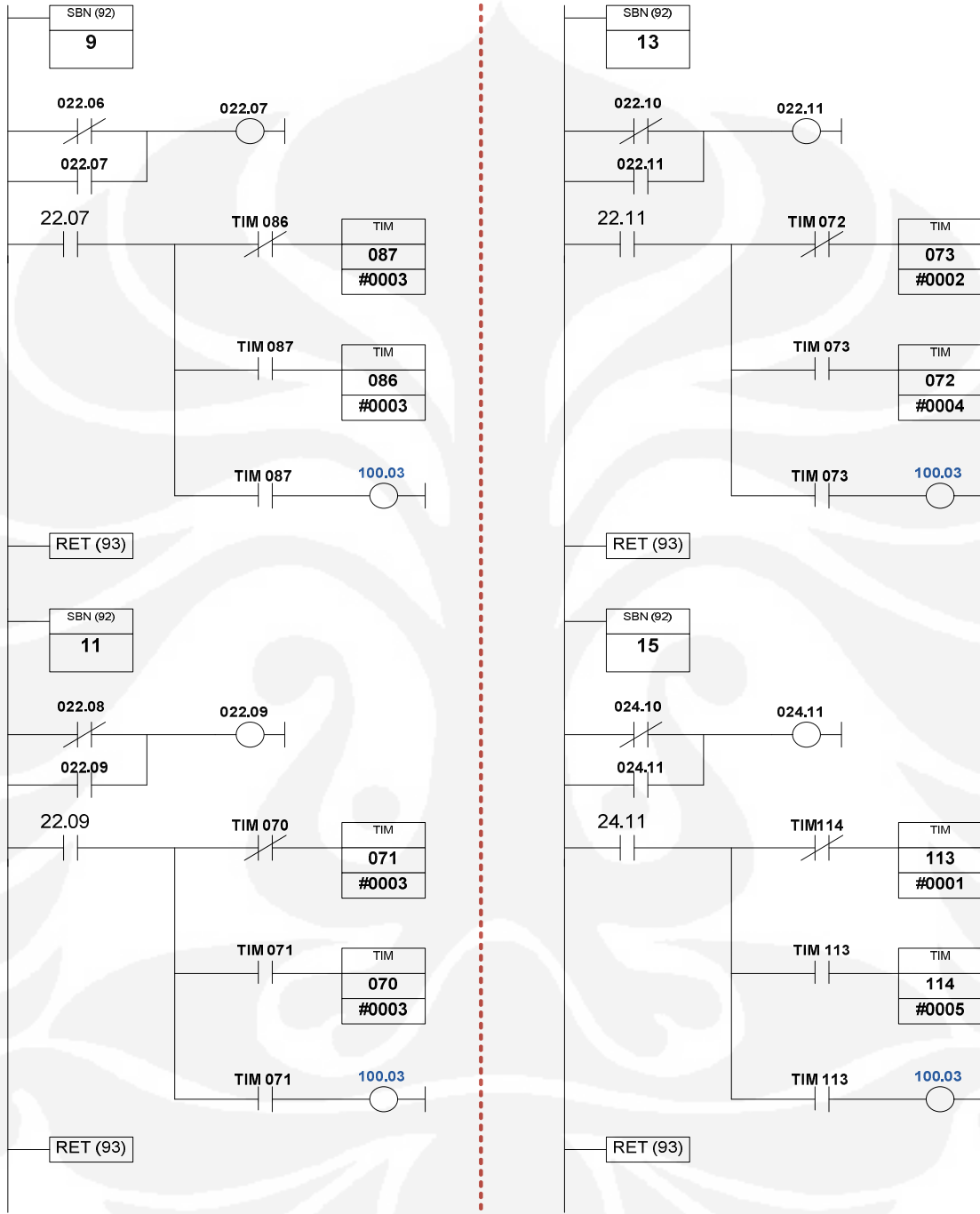
LAMPIRAN H

PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 9

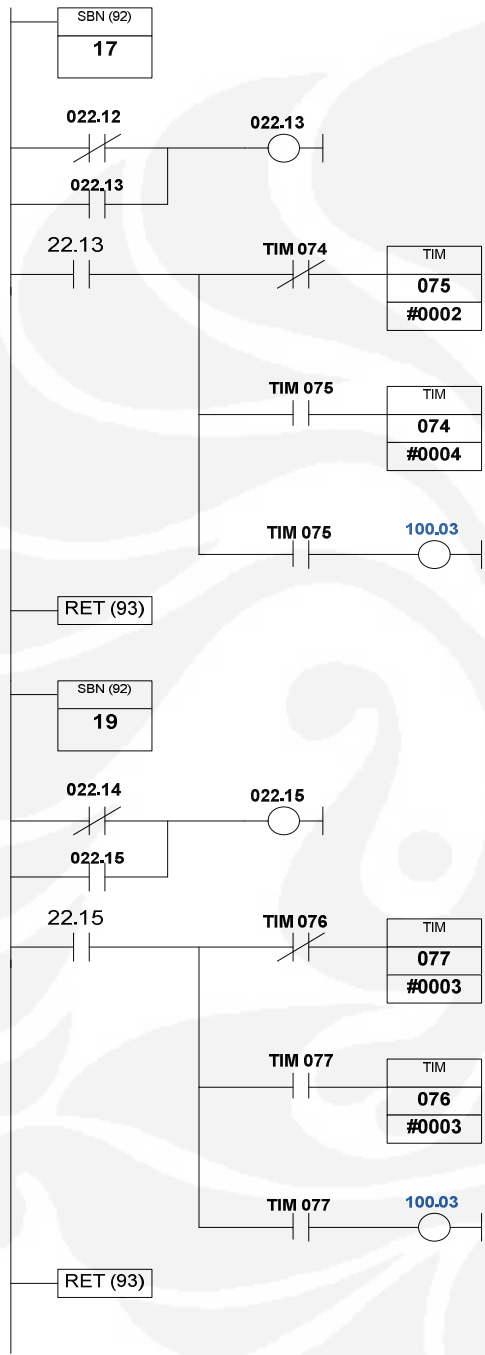


LAMPIRAN I

PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 10

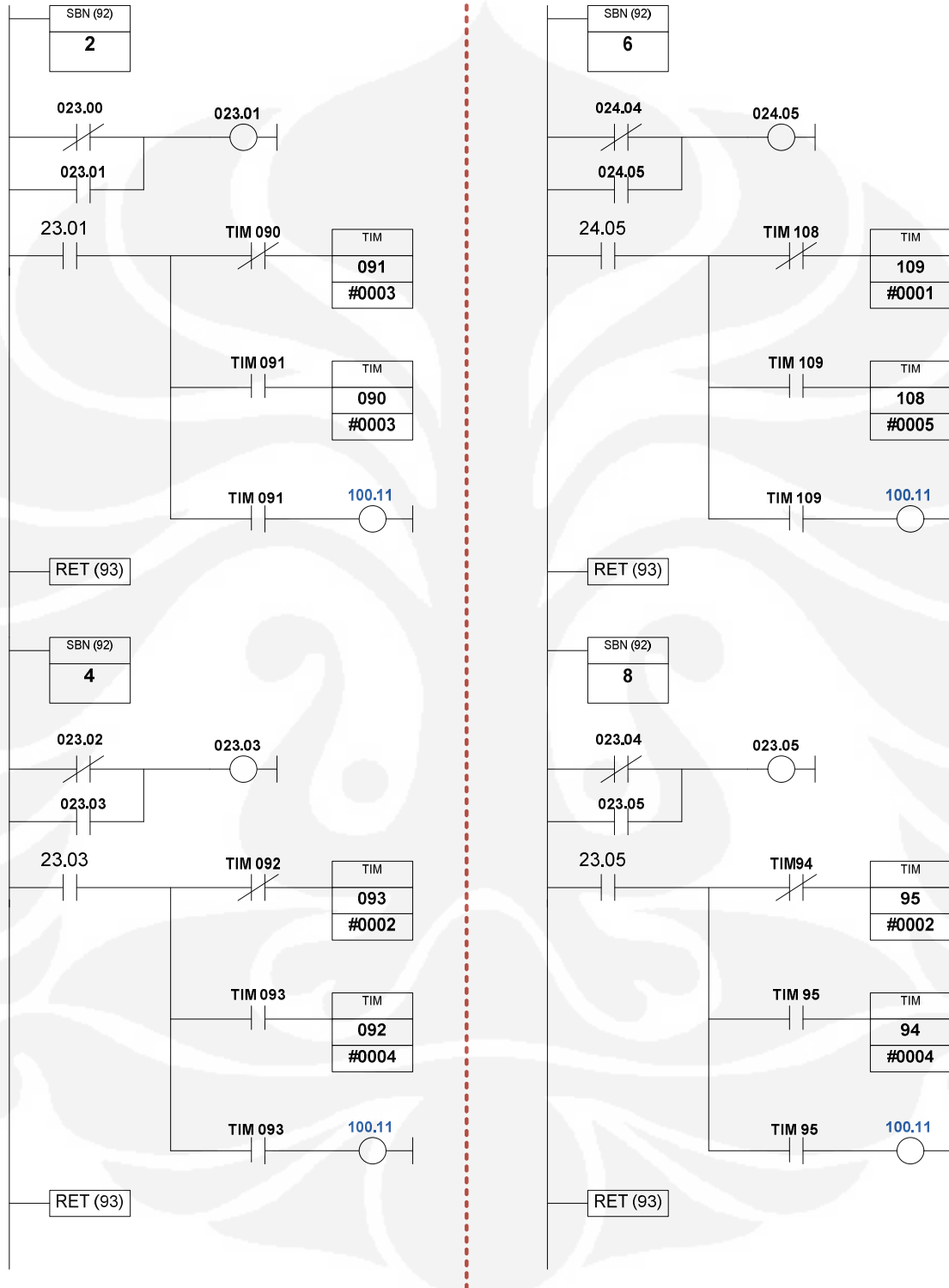


LAMPIRAN J
PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 11

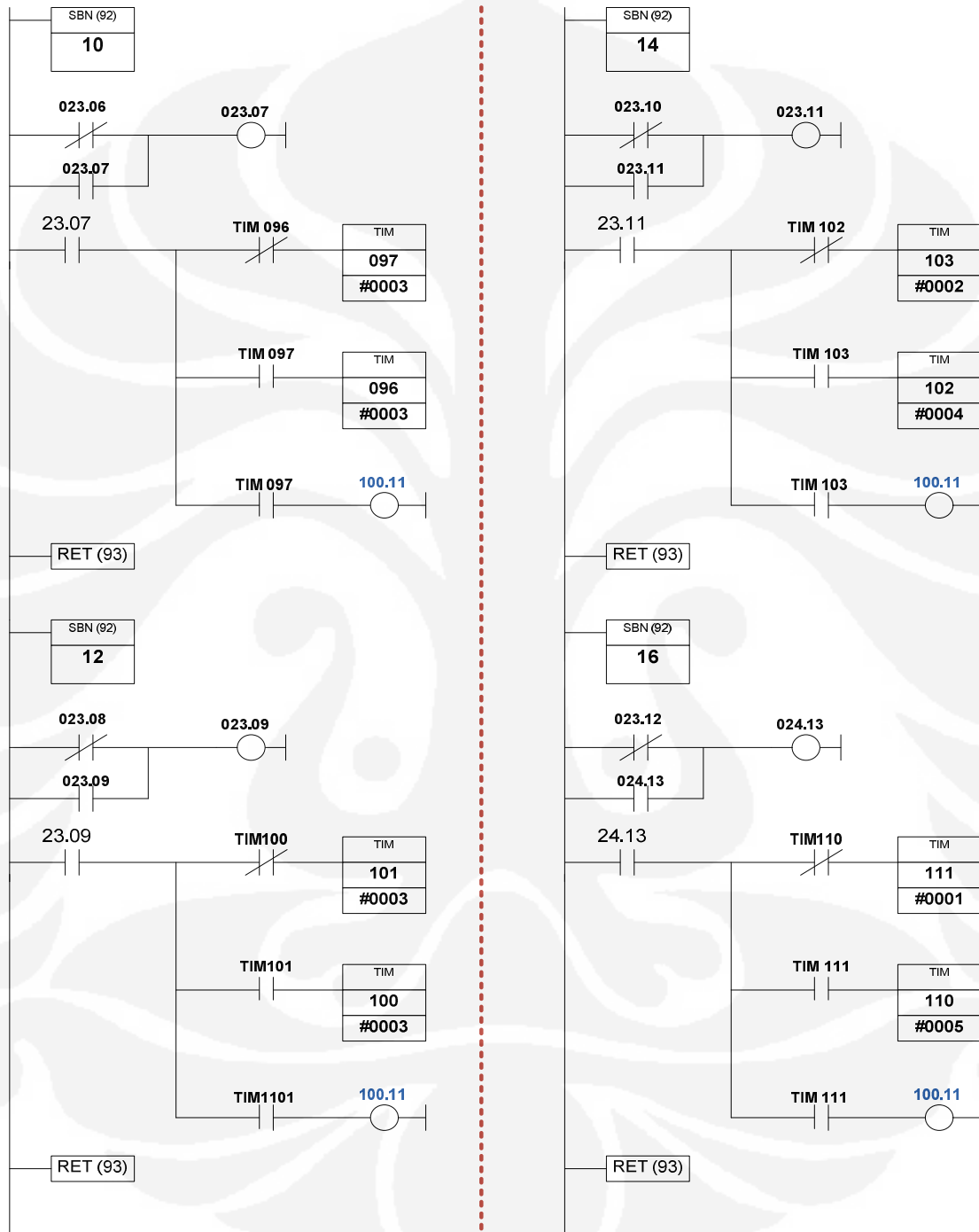


LAMPIRAN K

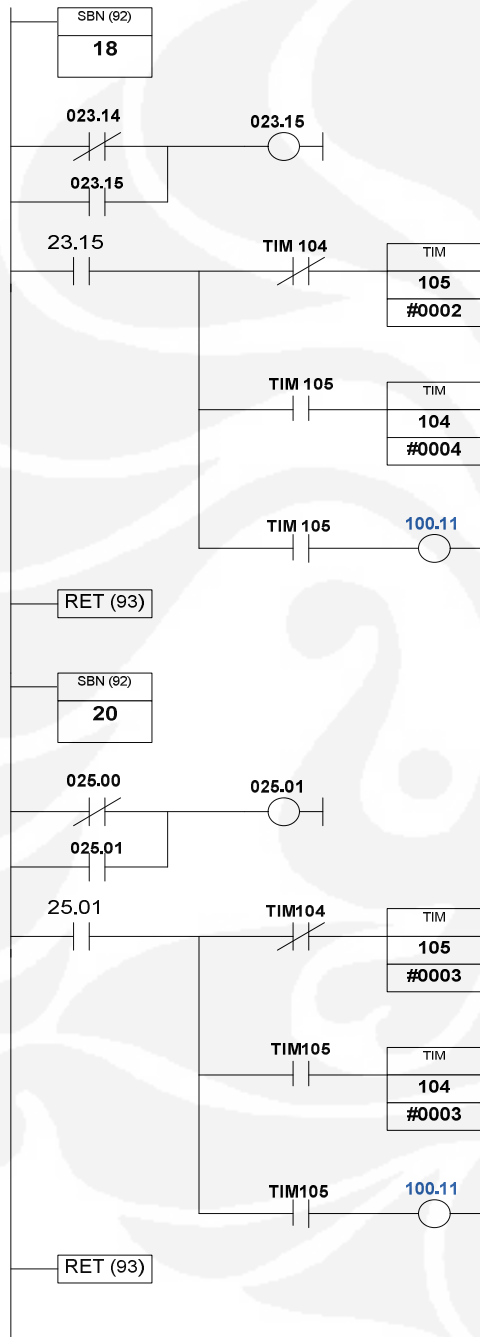
PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 12



LAMPIRAN L
PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 13



LAMPIRAN M
PEMROGRAMAN LADDER SYSWIN 3.4 BAGIAN 14



LAMPIRAN N

MAIN CODE SISTEM MONITORING HMI BAGIAN 1

<pre>Dim BENAR, BENAR2 As Boolean Dim DATA_IN, DATA_IN2 As String Dim DAYS, MACHINE As String Dim DAYS2, MACHINEB As String Private Sub Form_Load() Call CMDSTART_Click Call CMDSTART2_Click Call INTRODUCTION CMBTYPE.AddItem "FIBER" CMBTYPE.AddItem "PCP" CMBTYPE2.AddItem "FIBER" CMBTYPE2.AddItem "PCP" CMBMACHINE.AddItem "MACHINE1" CMBMACHINEB.AddItem "MACHINE2" CMBDAYS.AddItem "MONDAY" CMBDAYS.AddItem "TUESDAY" CMBDAYS.AddItem "WEDNESDAY" CMBDAYS.AddItem "THURSDAY" CMBDAYS.AddItem "FRIDAY" CMBDAYS2.AddItem "MONDAY" CMBDAYS2.AddItem "TUESDAY" CMBDAYS2.AddItem "WEDNESDAY" CMBDAYS2.AddItem "THURSDAY" CMBDAYS2.AddItem "FRIDAY" BENAR = False BENAR2 = False DATA_IN = "" DATA_IN2 = "" Form1.Caption = "HUMAN MACHINE INTERFACE" Image1.Visible = False Image2.Visible = False Image3.Visible = False Image4.Visible = False Image5.Visible = False Image6.Visible = False Image7.Visible = False Image8.Visible = False Image9.Visible = False Image10.Visible = False End Sub</pre>	<pre>Private Sub MSComm1_OnComm() Dim inp As Integer Select Case MSComm1.CommEvent Case comEvReceive If MSComm1.InBufferCount >= 0 Then BUFFER = CStr(MSComm1.Input) Text1.Text = BUFFER inp = Val(BUFFER) If inp > 7 Then 'ada 8 bit inp = 0 Else DATA_IN = DATA_IN & BUFFER Text1.Text = "DATA_VAL=" & DATA_IN ----- If MACHINE = "MACHINE1" Then If DAYS = "MONDAY" Then LPMONDAY.Caption = "C." & BUFFER Timer1.Enabled = 1 Elseif DAYS = "TUESDAY" Then LPTUESDAY.Caption = "C." & BUFFER Timer2.Enabled = 1 Elseif DAYS = "WEDNESDAY" Then LPWEDNESDAY.Caption = "C." & BUFFER Timer3.Enabled = 1 Elseif DAYS = "THURSDAY" Then LPTHURSDAY.Caption = "C." & BUFFER Timer4.Enabled = 1 Elseif DAYS = "FRIDAY" Then LPFRIDAY.Caption = "C." & BUFFER Timer5.Enabled = 1 End If If Len(DATA_IN) = 1 Then Call VALIDASI11 Elseif Len(DATA_IN) = 2 Then Call VALIDASI12 Elseif Len(DATA_IN) = 3 Then Call VALIDASI13 Elseif Len(DATA_IN) = 4 Then Call VALIDASI14 Elseif Len(DATA_IN) = 5 Then Call VALIDASI15 End If End If End If End If End Select Exit Sub End Sub</pre>
---	---

PAGE 1

PAGE 2

LAMPIRAN 0

MAIN CODE SISTEM MONITORING HMI BAGIAN 2

```
Private Sub MSComm2_OnComm()  
Dim inpB As Integer  
Select Case MSComm2.CommEvent  
Case comEvReceive  
If MSComm2.InBufferCount >= 0 Then  
BUFFER2 = CStr(MSComm2.Input)  
Text2.Text = BUFFER2  
  
inpB = Val(BUFFER2)  
If inpB > 7 Then 'ada 8 bit  
inpB = 0  
Else  
  
DATA_IN2 = DATA_IN2 & BUFFER2  
Text2.Text = "DATA_VAL=" & DATA_IN2  
-----  
If MACHINEB = "MACHINE2" Then  
  
If DAYS = "MONDAY" Then  
MPMONDAY.Caption = "C." & BUFFER2  
Timer6.Enabled = 1  
Elseif DAYS = "TUESDAY" Then  
MPTUESDAY.Caption = "C." & BUFFER2  
Timer7.Enabled = 1  
Elseif DAYS = "WEDNESDAY" Then  
MPWEDNESDAY.Caption = "C." & BUFFER2  
Timer8.Enabled = 1  
Elseif DAYS = "THURSDAY" Then  
MPTHURSDAY.Caption = "C." & BUFFER2  
Timer9.Enabled = 1  
Elseif DAYS = "FRIDAY" Then  
MPFRIDAY.Caption = "C." & BUFFER2  
Timer10.Enabled = 1  
End If  
If Len(DATA_IN2) = 1 Then  
Call VALIDASI21  
Elseif Len(DATA_IN2) = 2 Then  
Call VALIDASI22  
Elseif Len(DATA_IN2) = 3 Then  
Call VALIDASI23  
Elseif Len(DATA_IN2) = 4 Then  
Call VALIDASI24  
Elseif Len(DATA_IN2) = 5 Then  
Call VALIDASI25  
End If  
End If  
End If  
End If  
End Select  
Exit Sub  
End Sub
```

PAGE 3

```
Private Sub CMDSTART_Click()  
For i = 0 To 7  
Next i  
If MSComm1.PortOpen = True Then  
MsgBox "AKTIF"  
Elseif MSComm1.PortOpen = False Then  
MSComm1.PortOpen = True  
MsgBox "MACHINE1"  
End If  
End Sub  
Private Sub CMDSTART2_Click()  
For J = 0 To 7  
Next J  
If MSComm2.PortOpen = True Then  
MsgBox "AKTIF2"  
Elseif MSComm2.PortOpen = False Then  
MSComm2.PortOpen = True  
MsgBox "MACHINE2"  
End If  
End Sub  
Private Sub CMBTYPE_Click()  
TYPEPCB = CMBTYPE.Text  
MsgBox TYPEPCB  
DATA_IN = ""  
End Sub  
Private Sub CMBTYPE2_Click()  
TYPEPCB = CMBTYPE2.Text  
MsgBox TYPEPCB  
DATA_IN = ""  
End Sub  
Private Sub CMBDAYS_Click()  
DAYS = CMBDAYS.Text  
MsgBox DAYS  
DATA_IN = ""  
End Sub  
Private Sub CMBDAYS2_Click()  
DAYS = CMBDAYS2.Text  
MsgBox DAYS  
DATA_IN2 = ""  
End Sub  
Private Sub CMBMACHINE_Click()  
MACHINE = CMBMACHINE.Text  
MsgBox MACHINE  
DATA_IN = ""  
End Sub  
Private Sub CMBMACHINEB_Click()  
MACHINEB = CMBMACHINEB.Text  
MsgBox MACHINEB  
DATA_IN2 = ""  
End Sub  
Private Sub CMDSTOP_Click()  
On Error Resume Next  
MSComm1.PortOpen = False  
End Sub  
Private Sub CMDSTOP2_Click()  
On Error Resume Next  
MSComm2.PortOpen = False  
End Sub
```

PAGE 4

LAMPIRAN P

MAIN CODE SISTEM MONITORING HMI BAGIAN 3

<pre>Sub VALIDASIBENAR() If MACHINE = "MACHINE1" Then If BENAR = True Then If Len(DATA_IN) >= 5 Then If DAYS = "MONDAY" Then LTMONDAY.Caption = Val(LTMONDAY.Caption) + 1 Call TimerA ElseIf DAYS = "TUESDAY" Then LTTUESDAY.Caption = Val(LTTUESDAY.Caption) + 1 Call TIMERB ElseIf DAYS = "WEDNESDAY" Then LTWEDNESDAY.Caption = Val(LTWEDNESDAY.Caption) + 1 Call TIMERC ElseIf DAYS = "THURSDAY" Then LTTHURSDAY.Caption = Val(LTTHURSDAY.Caption) + 1 Call TIMERD ElseIf DAYS = "FRIDAY" Then LTFRIDAY.Caption = Val(LTFRIDAY.Caption) + 1 Call TIMERE End If DATA_IN = "" End If Else DATA_IN = "" If DAYS = "MONDAY" Then LFMONDAY.Caption = Val(LFMONDAY.Caption) + 1 Call TimerA Call FDATAEXPORT ElseIf DAYS = "TUESDAY" Then LFTUESDAY.Caption = Val(LFTUESDAY.Caption) + 1 Call TIMERB Call FDATAEXPORT1 ElseIf DAYS = "WEDNESDAY" Then LFWEDNESDAY.Caption = Val(LFWEDNESDAY.Caption) + 1 Call TIMERC Call FDATAEXPORT2 ElseIf DAYS = "THURSDAY" Then LFTHURSDAY.Caption = Val(LFTHURSDAY.Caption) + 1 Call TIMERD Call FDATAEXPORT3 ElseIf DAYS = "FRIDAY" Then LFFRIDAY.Caption = Val(LFFRIDAY.Caption) + 1 Call TIMERE Call FDATAEXPORT4 End If End If '===== End If End Sub</pre>	<pre>Sub VALIDASIBENAR2() If MACHINEB = "MACHINE2" Then If BENAR2 = True Then If Len(DATA_IN2) >= 5 Then If DAYS = "MONDAY" Then MTMONDAY.Caption = Val(MTMONDAY.Caption) + 1 Call TIMERF ElseIf DAYS = "TUESDAY" Then MTTUESDAY.Caption = Val(MTTUESDAY.Caption) + 1 Call TIMERG ElseIf DAYS = "WEDNESDAY" Then MTWEDNESDAY.Caption = Val(MTWEDNESDAY.Caption) + 1 Call TIMERH ElseIf DAYS = "THURSDAY" Then MTTHURSDAY.Caption = Val(MTTHURSDAY.Caption) + 1 Call TIMERI ElseIf DAYS = "FRIDAY" Then MTFRIDAY.Caption = Val(MTFRIDAY.Caption) + 1 Call TIMERJ End If DATA_IN2 = "" End If Else DATA_IN2 = "" If DAYS = "MONDAY" Then MFMONDAY.Caption = Val(MFMONDAY.Caption) + 1 Call TIMERF Call FDATAEXPORTA ElseIf DAYS = "TUESDAY" Then MFTUESDAY.Caption = Val(MFTUESDAY.Caption) + 1 Call TIMERG Call FDATAEXPORTB ElseIf DAYS = "WEDNESDAY" Then MFWEDNESDAY.Caption = Val(MFWEDNESDAY.Caption) + 1 Call TIMERH Call FDATAEXPORTC ElseIf DAYS = "THURSDAY" Then MFTHURSDAY.Caption = Val(MFTHURSDAY.Caption) + 1 Call TIMERI Call FDATAEXPORTD ElseIf DAYS = "FRIDAY" Then MFFRIDAY.Caption = Val(MFFRIDAY.Caption) + 1 Call TIMERJ Call FDATAEXPORTE End If End If '===== End If End Sub</pre>
---	--

PAGE 5

PAGE 6

LAMPIRAN Q

MAIN CODE SISTEM MONITORING HMI BAGIAN 4

<pre>Sub VALIDASISALAH() DATA_IN = "" If DAYS = "MONDAY" Then LFMONDAY.Caption = Val(LFMONDAY.Caption) + 1 Call TimerA Call FDATAEXPORT ElseIf DAYS = "TUESDAY" Then LFTUESDAY.Caption = Val(LFTUESDAY.Caption) + 1 Call TIMERB Call FDATAEXPORT1 ElseIf DAYS = "WEDNESDAY" Then LFWEDNESDAY.Caption = Val(LFWEDNESDAY.Caption) + 1 Call TIMERC Call FDATAEXPORT2 ElseIf DAYS = "THURSDAY" Then LFTHURSDAY.Caption = Val(LFTHURSDAY.Caption) + 1 Call TIMERD Call FDATAEXPORT3 ElseIf DAYS = "FRIDAY" Then LFFRIDAY.Caption = Val(LFFRIDAY.Caption) + 1 Call TIMERE Call FDATAEXPORT4 End If End If End Sub Sub VALIDASISALAH2() DATA_IN2 = "" If DAYS = "MONDAY" Then MFMONDAY.Caption = Val(MFMONDAY.Caption) + 1 Call TIMERF Call FDATAEXPORTA ElseIf DAYS = "TUESDAY" Then MFTUESDAY.Caption = Val(MFTUESDAY.Caption) + 1 Call TIMERG Call FDATAEXPORTB ElseIf DAYS = "WEDNESDAY" Then MFWEDNESDAY.Caption = Val(MFWEDNESDAY.Caption) + 1 Call TIMERH Call FDATAEXPORTC ElseIf DAYS = "THURSDAY" Then MFTHURSDAY.Caption = Val(MFTHURSDAY.Caption) + 1 Call FDATAEXPORTD ElseIf DAYS = "FRIDAY" Then MFFRIDAY.Caption = Val(MFFRIDAY.Caption) + 1 Call FDATAEXPORTE End If End If End Sub</pre>	<pre>Sub VALIDASI11() If Left(DATA_IN, 1) = "0" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE0 Call simulation End Sub Sub VALIDASI12() If Mid(DATA_IN, 2, 1) = "1" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE1 Call simulation End Sub Sub VALIDASI13() If Mid(DATA_IN, 3, 1) = "2" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE2 Call simulation1 End Sub Sub VALIDASI14() If Mid(DATA_IN, 4, 1) = "1" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE1 Call simulation1 End Sub Sub VALIDASI15() If Left(DATA_IN, 1) = "0" Then BENAR = True Else BENAR = False End If Call VALIDASIBENAR Call DUTYCYCLE0 Timer11.Enabled = True Call simulation2 End Sub Sub VALIDASI16() If Val(DATA_IN, 6) = "6" Then BENAR = True Else BENAR = False End If Call VALIDASISALAH End Sub</pre>
--	---

PAGE 7

PAGE 8

LAMPIRAN R

MAIN CODE SISTEM MONITORING HMI BAGIAN 5

<pre>Sub VALIDASI21() If Left(DATA_IN2, 1) = "0" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call simulationa Call DUTYCYCLEA End Sub Sub VALIDASI22() If Mid(DATA_IN2, 2, 1) = "1" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call DUTYCYCLEB Call simulationa End Sub Sub VALIDASI23() If Mid(DATA_IN2, 3, 1) = "2" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call simulationb Call DUTYCYCLEC End Sub Sub VALIDASI24() If Mid(DATA_IN2, 4, 1) = "1" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call simulationb Call DUTYCYCLEB End Sub Sub VALIDASI25() If Left(DATA_IN2, 1) = "0" Then BENAR2 = True Else BENAR2 = False End If Call VALIDASIBENAR2 Call simulationc Call DUTYCYCLEA Timer12.Enabled = True End Sub Sub VALIDASI26() If Val(DATA_IN, 6) = "6" Then BENAR = True Else BENAR = False End If Call VALIDASISALAH2 End Sub</pre>	<pre>Private Sub RECORD_Click() SPREADSHEET1.Visible = True End Sub Private Sub DRECORD_Click() SPREADSHEET1.Visible = False End Sub '=====TimeprocessM1===== Private Sub Timer1_timer() LTPMONDAY.Caption = Val(LTPMONDAY.Caption) + 1 End Sub Private Sub Timer2_timer() LPTTUESDAY.Caption = Val(LPTTUESDAY.Caption) + 1 End Sub Private Sub Timer3_timer() LTPWEDNESDAY.Caption = Val(LTPWEDNESDAY.Caption) + 1 End Sub Private Sub Timer4_timer() LPTTHURSDAY.Caption = Val(LPTTHURSDAY.Caption) + 1 End Sub Private Sub Timer5_timer() LTPFRIDAY.Caption = Val(LTPFRIDAY.Caption) + 1 End Sub Private Sub TimerA() LTPMONDAY.Caption = 0 Timer1.Enabled = 0 TimerLD1 = True TimerLD1X = True Call DATAEXPORT End Sub Private Sub TIMERB() LPTTUESDAY.Caption = 0 Timer2.Enabled = 0 TimerLD2 = True TimerLD2X = True Call DATAEXPORT1 End Sub Private Sub TIMERC() LTPWEDNESDAY.Caption = 0 Timer3.Enabled = 0 TimerLD3 = True TimerLD3X = True Call DATAEXPORT2 End Sub Private Sub TIMERD() LPTTHURSDAY.Caption = 0 Timer4.Enabled = 0 TimerLD4 = True TimerLD4X = True Call DATAEXPORT3 End Sub Private Sub TIMERE() LTPFRIDAY.Caption = 0 Timer5.Enabled = 0 TimerLD5 = True TimerLD5X = True Call DATAEXPORT4 End Sub</pre>
--	---

PAGE 9

PAGE 10

LAMPIRAN S

MAIN CODE SISTEM MONITORING HMI BAGIAN 6

<pre>'=====TIME PROCESSM2===== Private Sub Timer6_Timer() MTPMONDAY.Caption = Val(MTPMONDAY.Caption) + 1 End Sub Private Sub Timer7_Timer() MTPTUESDAY.Caption = Val(MTPTUESDAY.Caption) + 1 End Sub Private Sub Timer8_Timer() MTPWEDNESDAY.Caption = Val(MTPWEDNESDAY.Caption) + 1 End Sub Private Sub Timer9_Timer() MTPTHURSDAY.Caption = Val(MTPTHURSDAY.Caption) + 1 End Sub Private Sub Timer10_Timer() MTPFRIDAY.Caption = Val(MTPFRIDAY.Caption) + 1 End Sub Private Sub TIMERF() MTPMONDAY.Caption = 0 Timer6.Enabled = False TimerLDA = True TimerLDAX = True DATAEXPORTA End Sub Private Sub TIMERG() MTPTUESDAY.Caption = 0 Timer7.Enabled = False TimerLDB = True TimerLDBX = True DATAEXPORTB End Sub Private Sub TIMERH() MTPWEDNESDAY.Caption = 0 Timer8.Enabled = False TimerLDC = True TimerLDCX = True DATAEXPORTC End Sub Private Sub TIMERI() MTPTHURSDAY.Caption = 0 Timer9.Enabled = False TimerLDD = True TimerLDDX = True DATAEXPORTD End Sub Private Sub TIMERJ() MTPFRIDAY.Caption = 0 Timer10.Enabled = False TimerLDE = True TimerLDEX = True DATAEXPORTE End Sub</pre>	<pre>'-----LASTDELAYM1----- Private Sub TimerLD1_Timer() LDMONDAY.Caption = Val(LDMONDAY.Caption) - 1 End Sub Private Sub TimerLD1X_Timer() TimerLD1 = False TimerLD1X = False LDMONDAY.Caption = 10 End Sub Private Sub TimerLD2_Timer() LDTUESDAY.Caption = Val(LDTUESDAY.Caption) - 1 End Sub Private Sub TimerLD2X_Timer() TimerLD2 = False TimerLD2X = False LDTUESDAY.Caption = 10 End Sub Private Sub TimerLD3_Timer() LDWEDNESDAY.Caption = Val(LDWEDNESDAY.Caption) - 1 End Sub Private Sub TimerLD3X_Timer() TimerLD3 = False TimerLD3X = False LDWEDNESDAY.Caption = 10 End Sub Private Sub TimerLD4_Timer() LDTHURSDAY.Caption = Val(LDTHURSDAY.Caption) - 1 End Sub Private Sub TimerLD4X_Timer() TimerLD4 = False TimerLD4X = False LDTHURSDAY.Caption = 10 End Sub Private Sub TimerLD5_Timer() LDFRIDAY.Caption = Val(LDFRIDAY.Caption) - 1 End Sub Private Sub TimerLD5X_Timer() TimerLD5 = False TimerLD5X = False LDFRIDAY.Caption = 10 End Sub</pre>
---	--

PAGE 11

PAGE 12

LAMPIRAN T

MAIN CODE SISTEM MONITORING HMI BAGIAN 7

<pre>'-----LASTDELAYM2----- Private Sub TimerLDA_Timer() LD2MONDAY.Caption = Val(LD2MONDAY.Caption) - 1 End Sub Private Sub TimerLDAX_Timer() TimerLDA = False TimerLDAX = False LD2MONDAY.Caption = 10 End Sub Private Sub TimerLDB_Timer() LD2TUESDAY.Caption = Val(LD2TUESDAY.Caption) - 1 End Sub Private Sub TimerLDBX_Timer() TimerLDB = False TimerLDBX = False LD2TUESDAY.Caption = 10 End Sub Private Sub TimerLDC_Timer() LD2WEDNESDAY.Caption = Val(LD2WEDNESDAY.Caption) - 1 End Sub Private Sub TimerLDCX_Timer() TimerLDC = False TimerLDCX = False LD2WEDNESDAY.Caption = 10 End Sub Private Sub TimerLDD_Timer() LD2THURSDAY.Caption = Val(LD2THURSDAY.Caption) - 1 End Sub Private Sub TimerLDDX_Timer() TimerLDD = False TimerLDDX = False LD2THURSDAY.Caption = 10 End Sub Private Sub TimerLDE_Timer() LD2FRIDAY.Caption = Val(LD2FRIDAY.Caption) - 1 End Sub Private Sub TimerLDEX_Timer() TimerLDE = False TimerLDEX = False LD2FRIDAY.Caption = 10 End Sub '-----INTRODUCTION----- Private Sub INTRODUCTION() Picture15.Visible = True Picture16.Visible = False Picture17.Visible = False Picture18.Visible = False TimerIM1 = True End Sub Private Sub TimerIM1_Timer() Picture15.Visible = False Picture16.Visible = True Picture17.Visible = False Picture18.Visible = False TimerIM1 = False TimerIM2 = True End Sub</pre>	<pre>Private Sub TimerIM2_Timer() Picture15.Visible = False Picture16.Visible = False Picture17.Visible = True Picture18.Visible = False TimerIM2 = False TimerIM3 = True End Sub Private Sub TimerIM3_Timer() Picture15.Visible = False Picture16.Visible = False Picture17.Visible = False Picture18.Visible = True TimerIM3 = False TimerIM4 = True End Sub Private Sub TimerIM4_Timer() Picture15.Visible = False Picture16.Visible = False Picture17.Visible = True Picture18.Visible = False TimerIM4 = False TimerIM5 = True End Sub Private Sub TimerIM5_Timer() Picture15.Visible = False Picture16.Visible = True Picture17.Visible = False Picture18.Visible = False TimerIM5 = False TimerIM6 = True End Sub Private Sub TimerIM6_Timer() TimerIM6 = False Call INTRODUCTION End Sub '-----calendar----- Private Sub timkalender_Timer() Dim Hari_ini As Variant Hari_ini = Now Ibhari.Caption = Format(Hari_ini, "dddd") Ibbulan.Caption = Format(Hari_ini, "mmmm") Ibtahun.Caption = Format(Hari_ini, "yyyy") Iblangka.Caption = Format(Hari_ini, "d") Ibwaktu.Caption = Format(Hari_ini, "hh:mm:ss") End Sub</pre>
--	--

PAGE 13

PAGE 14

LAMPIRAN U

MAIN CODE SISTEM MONITORING HMI BAGIAN 8

<pre> '-----DATAEXPORTM1----- Private Sub DATAEXPORT() SPREADSHEET1.Cells(4, 4) = LTMONDAY.Caption End Sub Private Sub DATAEXPORT1() SPREADSHEET1.Cells(4, 5) = LTTUESDAY.Caption End Sub Private Sub DATAEXPORT2() SPREADSHEET1.Cells(4, 6) = LTWEDNESDAY.Caption End Sub Private Sub DATAEXPORT3() SPREADSHEET1.Cells(4, 7) = LTTTHURSDAY.Caption End Sub Private Sub DATAEXPORT4() SPREADSHEET1.Cells(4, 8) = LTFRIDAY.Caption End Sub Private Sub FDATAEXPORT() SPREADSHEET1.Cells(5, 4) = LFMONDAY.Caption End Sub Private Sub FDATAEXPORT1() SPREADSHEET1.Cells(5, 5) = LFTUESDAY.Caption End Sub Private Sub FDATAEXPORT2() SPREADSHEET1.Cells(5, 6) = LFWEDNESDAY.Caption End Sub Private Sub FDATAEXPORT3() SPREADSHEET1.Cells(5, 7) = LFTTHURSDAY.Caption End Sub Private Sub FDATAEXPORT4() SPREADSHEET1.Cells(5, 8) = LFWEDNESDAY.Caption End Sub '-----DATAEXPORTM2----- Private Sub DATAEXPORTA() SPREADSHEET1.Cells(9, 4) = MTMONDAY.Caption End Sub Private Sub DATAEXPORTB() SPREADSHEET1.Cells(9, 5) = MTTUESDAY.Caption End Sub Private Sub DATAEXPORTC() SPREADSHEET1.Cells(9, 6) = MTWEDNESDAY.Caption End Sub Private Sub DATAEXPORTD() SPREADSHEET1.Cells(9, 7) = MTTTHURSDAY.Caption End Sub Private Sub DATAEXPORTE() SPREADSHEET1.Cells(9, 8) = MTFRIDAY.Caption End Sub Private Sub FDATAEXPORTA() SPREADSHEET1.Cells(10, 4) = MFMONDAY.Caption End Sub Private Sub FDATAEXPORTB() SPREADSHEET1.Cells(10, 5) = MFTUESDAY.Caption End Sub Private Sub FDATAEXPORTC() SPREADSHEET1.Cells(10, 6) = MFWEDNESDAY.Caption End Sub Private Sub FDATAEXPORTD() SPREADSHEET1.Cells(10, 7) = MFTTHURSDAY.Caption End Sub Private Sub FDATAEXPORTE() SPREADSHEET1.Cells(10, 8) = MFFRIDAY.Caption End Sub </pre>	<pre> '-----DUTYCYCLEM1----- Private Sub DUTYCYCLE0() Label29.Visible = True Label32.Visible = False Label33.Visible = False End Sub Private Sub DUTYCYCLE1() Label29.Visible = False Label32.Visible = True Label33.Visible = False End Sub Private Sub DUTYCYCLE2() Label29.Visible = False Label32.Visible = False Label33.Visible = True End Sub Private Sub Timer11_Timer() Label29.Visible = False Timer11.Enabled = False End Sub '-----DUTYCYCLEM2----- Private Sub DUTYCYCLEA() Label34.Visible = True Label35.Visible = False Label36.Visible = False End Sub Private Sub DUTYCYCLEB() Label34.Visible = False Label35.Visible = True Label36.Visible = False End Sub Private Sub DUTYCYCLEC() Label34.Visible = False Label35.Visible = False Label36.Visible = True End Sub Private Sub Timer12_Timer() Label34.Visible = False Timer12.Enabled = False End Sub </pre>
PAGE 15	PAGE 16

LAMPIRAN V

MAIN CODE SISTEM MONITORING HMI BAGIAN 9

<pre>'-----simulationm1----- Private Sub simulation() Image5.Visible = False Image3.Visible = False Image4.Visible = False Timer15.Enabled = False Timer16.Enabled = False Timer17.Enabled = False Timer18.Enabled = False Image1.Visible = True Image2.Visible = False Timer13.Enabled = True End Sub Private Sub Timer13_Timer() Image1.Visible = False Image2.Visible = True Timer13.Enabled = False Timer14.Enabled = True End Sub Private Sub Timer14_Timer() Timer14.Enabled = False Call simulation End Sub Private Sub simulation1() Image5.Visible = False Timer17.Enabled = False Timer18.Enabled = False Image1.Visible = False Image2.Visible = False Timer13.Enabled = False Timer14.Enabled = False Image3.Visible = True Image4.Visible = False Timer15.Enabled = True End Sub Private Sub Timer15_Timer() Image3.Visible = False Image4.Visible = True Timer15.Enabled = False Timer16.Enabled = True End Sub Private Sub Timer16_Timer() Timer16.Enabled = False Call simulation1 End Sub Private Sub simulation2() Image3.Visible = False Image4.Visible = False Timer15.Enabled = False Timer16.Enabled = False Image5.Visible = True Timer17.Enabled = True End Sub Private Sub Timer17_Timer() Image5.Visible = False Timer17.Enabled = False Timer18.Enabled = True End Sub Private Sub Timer18_Timer() Timer18.Enabled = False Call simulation2 End Sub</pre>	<pre>'-----simulationm2----- Private Sub simulationa() Image10.Visible = False Image8.Visible = False Image9.Visible = False Timer24.Enabled = False Timer23.Enabled = False Timer22.Enabled = False Timer21.Enabled = False Image6.Visible = True Image7.Visible = False Timer19.Enabled = True End Sub Private Sub Timer19_Timer() Image6.Visible = False Image7.Visible = True Timer19.Enabled = False Timer20.Enabled = True End Sub Private Sub Timer20_Timer() Timer20.Enabled = False Call simulationa End Sub Private Sub simulationb() Image10.Visible = False Timer24.Enabled = False Timer23.Enabled = False Image6.Visible = False Image7.Visible = False Timer19.Enabled = False Timer20.Enabled = False Image8.Visible = True Image9.Visible = False Timer21.Enabled = True End Sub Private Sub Timer21_Timer() Image9.Visible = True Image8.Visible = False Timer21.Enabled = False Timer22.Enabled = True End Sub Private Sub Timer22_Timer() Timer22.Enabled = False Call simulationb End Sub Private Sub simulationc() Image8.Visible = False Image9.Visible = False Timer21.Enabled = False Timer22.Enabled = False Image10.Visible = True Timer23.Enabled = True End Sub Private Sub Timer23_Timer() Image10.Visible = False Timer23.Enabled = False Timer24.Enabled = True End Sub Private Sub Timer24_Timer() Timer24.Enabled = False Call simulationc End Sub</pre>
--	--

PAGE 17

PAGE 18

LAMPIRAN W

DATASHEET MOTOR DC



TG-38243000 Series

(24VDC LOWNOISE LONG LIFE)

OUTPUT POWER: 0.8W(APPROX)

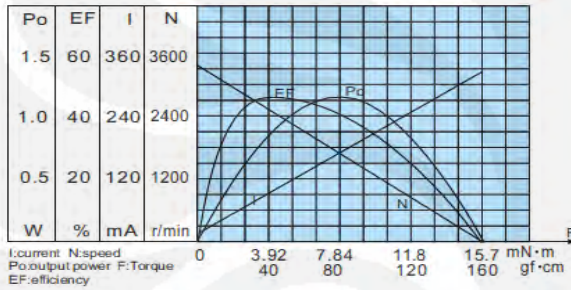
SGMADA[®]

DC GEARED MOTOR

WEIGHT:180~200g(APPROX)

Typical applications: Label printers, auto shutter, welding machines, water meter IC card, grill, oven, cleaning machine, garbage disposers, household appliances, slot machines, money detector, automatic actuator, coffee machine, towel disposal, lighting, coin refund devices, peristaltic pump.

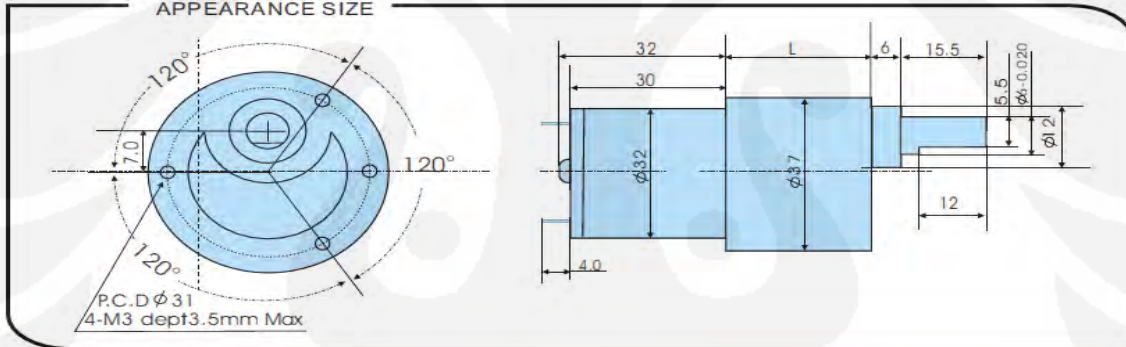
MOTOR CHARACTERISTICS



MOTOR TORQUE/SPEED/CURRENT

Rated voltage	24VDC
No load speed	3000r/min
No load current	18mA
Rated torque	40gf·cm 3.92mN·m
Rated current	85mA
Rated speed	2500r/min
Stall torque	160gf·cm 15.7mN·m
Stall current	300mA

APPEARANCE SIZE



GEARED MOTOR TORQUE/SPEED/CURRENT

Geared motor name	Rated Volt. V	No load		Load torque				At maximum efficiency		Output power W	Number of gear trains	Gearbox length "L" mm
		Current mA	Speed r/min	Current mA	Speed r/min	Torque		Kg·cm	N·m			
						Kg·cm	N·m					
TG-38243000-30K	24	≤18	100	≤85	83	0.88	0.09	3	0.29	0.73	3	22
TG-38243000-60K	24	≤18	50	≤85	41	1.58	0.15	6	0.59	0.65	4	25
TG-38243000-90K	24	≤18	33	≤85	28	2.38	0.23	7.0	0.69	0.65	4	25
TG-38243000-180K	24	≤18	16.5	≤85	14	4.25	0.42	13	1.27	0.60	5	27
TG-38243000-270K	24	≤18	11	≤85	9.2	6.37	0.62	20	1.96	0.60	5	27
TG-38243000-540K	24	≤18	5.5	≤85	5.0	10	0.98	30	2.94	0.5	6	29
TG-38243000-810K	24	≤18	3.7	≤50	3.4	10	0.98	30	2.94	0.34	6	29
TG-38243000-1620K	24	≤18	1.8	≤30	1.6	10	0.98	30	2.94	0.16	7	29
TG-38243000-2430K	24	≤18	1.2	≤25	1.1	10	0.98	30	2.94	0.11	7	29
Packing Details	CTN size:37X27XH18CM 60pcs/CTN N.W. 12Kgs											





