

**PENGAKSESAN MOTOR TIGA FASA MELALUI *INVERTER*  
YANG DIKONTROL DENGAN PLC**

**KERJA PRAKTEK**



Oleh :

**Daniel Kristianto                      07.41020.0017**

**Satiti Dewi                                07.41020.0022**

**SEKOLAH TINGGI  
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER  
SURABAYA**

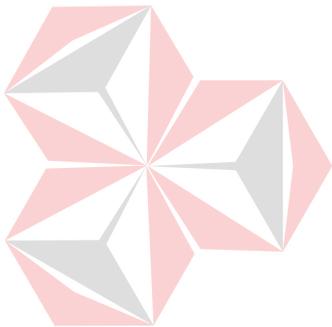
**2010**

**LAPORAN KERJA PRAKTEK**

**PENGAKSESAN MOTOR TIGA FASA MELALUI *INVERTER* YANG  
DIKONTROL DENGAN PLC**

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



Oleh :

Daniel Kristianto 07.41020.0017

Satiti Dewi 07.41020.0022

S1 (Strata Satu) Sistem Komputer

**SEKOLAH TINGGI**

**MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER**

**SURABAYA**

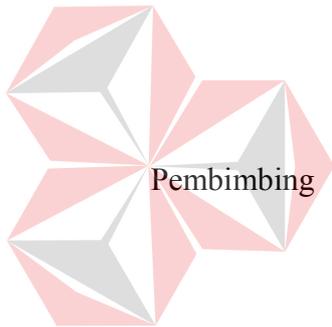
**2010**

## LAPORAN KERJA PRAKTEK

### PENGAKSESAN MOTOR TIGA FASA MELALUI *INVERTER* YANG DIKONTROL DENGAN PLC

Telah diperiksa, diuji dan disetujui

Surabaya, Maret 2010



Disetujui :

UNIVERSITAS

Dinamika

Penyelia

Harianto, S.Kom.

NIDN : 0722087701

Pauladie Susanto, S.Kom.

NIDN : 0729047501

Mengetahui :

Kaprodi S1 Sistem Komputer

Pauladie Susanto, S.Kom.

NIDN : 0729047501

## ABSTRAKSI

Motor tiga fasa biasanya digunakan untuk menggerakkan *conveyor* yang ada di pabrik-pabrik ataupun di *home industry*. Oleh karena itu, kami membuat percobaan untuk menggerakkan motor tiga fasa yang ada di Laboratorium PLC (*Programmable Logic Control*) STIKOM Surabaya.

*Inverter* adalah peralatan elektronik yang melakukan konfersi arus dari arus searah ke arus bolak-balik. Tegangan dan frekuensi arus hasil konfersi tergantung dari transformator dan alat kontrol yang digunakan.

Dengan kemajuan teknologi membuat masyarakat menjadi melupakan teknologi lama, seperti bahasa pemrograman pada FESTO. Banyak orang yang menggunakan bahasa pemrograman *Ladder Diagram* dan bahasa pemrograman STL, sedangkan bahasa pemrograman *Function Chart* jarang sekali digunakan.

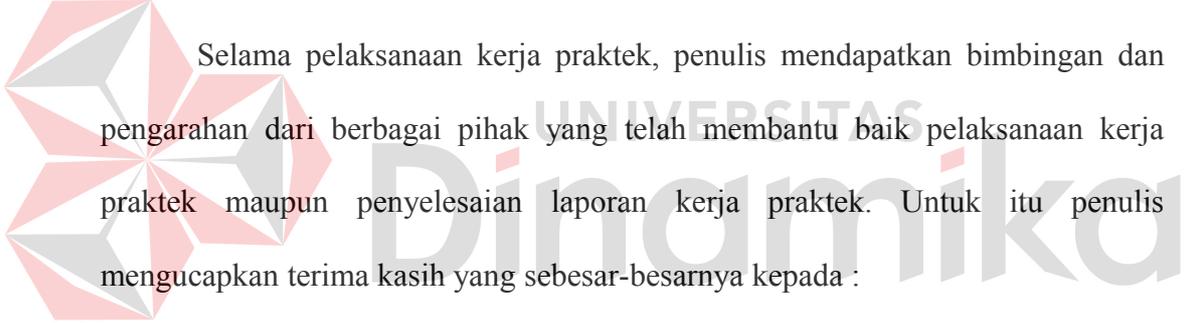
Oleh karena itu, kami menggunakan bahasa pemrograman *Function Chart* sebagai pengontrol menggunakan PLC yang digunakan untuk menggerakkan motor tiga fasa melalui *inverter* VF-S11.

Diharapkan dengan kami kerja praktek di Laboratorium PLC STIKOM Surabaya, kami dapat memberikan sesuatu yang bermanfaat bagi mahasiswa didalam menempuh perkuliahan maupun praktikum, agar para mahasiswa tidak hanya berbekal teori namun juga prakteknya.

## KATA PENGANTAR

Dengan rasa syukur kehadirat Tuhan Yang Maha Esa, penulis telah dapat menyelesaikan laporan kerja praktek yang merupakan pesyaratan dalam menyelesaikan Program Studi Strata Satu Sistem Komputer di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya (STIKOM Surabaya).

Laporan ini disusun sebagai bukti bahwa penulis telah menyelesaikan kerja praktek di Laboratorium PLC STIKOM Surabaya, selama 1 bulan semenjak tanggal 18 Januari 2010 sampai dengan 12 Februari 2010.

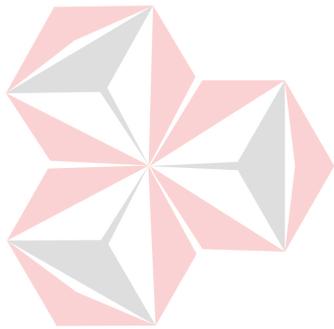


Selama pelaksanaan kerja praktek, penulis mendapatkan bimbingan dan pengarahan dari berbagai pihak yang telah membantu baik pelaksanaan kerja praktek maupun penyelesaian laporan kerja praktek. Untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Pauladie Susanto, S.Kom. selaku Kaprodi S1 Sistem Komputer yang sudah memberikan izin untuk melaksanakan kerja praktek di Laboratorium PLC STIKOM Surabaya.
2. Bapak Harianto, S.Kom.,M.Eng. sebagai dosen pembimbing kerja praktek.
3. Teman-teman penulis yang selalu siap memberikan semangat dan dorongan dalam penyelesaian kerja praktek ini.
4. Semua pihak yang tidak mungkin penulis sebutkan satu persatu disini, yang selama ini telah memberikan bantuan moral dan material kepada penulis.

Semoga Tuhan senantiasa memberikan rahmat dan perlindungannya kepada semua pihak yang telah banyak memberikan bantuan, bimbingan ataupun nasehat-nasehat dan bantuan moral baik secara langsung maupun tidak langsung kepada penulis.

Penulis sadar bahwa masih banyak kekurangan pada penulisan laporan kerja praktek ini. Namun penulis berharap semoga laporan kerja praktek ini dapat ikut menunjang perkembangan ilmu pengetahuan di masa yang akan datang.



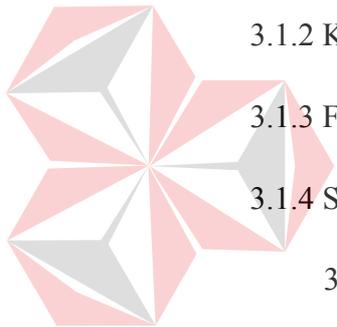
Surabaya, Maret 2010

UNIVERSITAS  
**Dinamika**  
Penulis

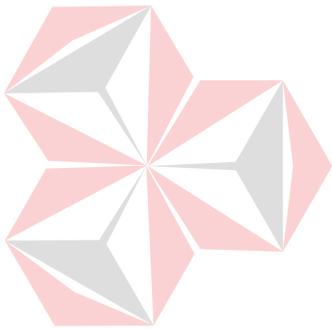
## DAFTAR ISI

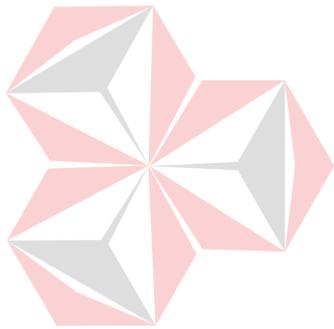
	Halaman
Halaman Judul .....	i
Lembar Pengesahan .....	ii
ABSTRAKSI .....	iii
KATA PENGANTAR .....	iv
DAFTAR ISI .....	vi
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR LAMPIRAN .....	xvii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Kontribusi .....	3
1.6 Sistematika Penulisan .....	3
BAB II GAMBARAN UMUM STIKOM SURABAYA .....	5
2.1 Sejarah dan Perkembangan .....	5
2.2 Lokasi .....	12
2.3 Jenis Usaha .....	12

2.4	Visi, Misi dan Komitmen STIKOM Surabaya .....		14
2.4.1	Visi	STIKOM	14
	Surabaya .....		14
2.4.2	Misi	STIKOM	15
	Surabaya .....		
2.4.3	Komitmen	STIKOM	17
	Surabaya .....		
BAB III LANDASAN TEORI .....			17
3.1	PLC .....		18
3.1.1	Sejarah .....		19
3.1.2	Konsep PLC .....		20
3.1.3	Fungsi PLC .....		21
3.1.4	Sistem Komponen PLC .....		21
3.1.4.1	CCU ( <i>Central Control Unit</i> ) .....		22
3.1.4.2	Modul Input / Output .....		23
3.1.4.3	<i>Input Device</i> .....		23
3.1.4.4	Actuator ( <i>Output Device</i> ) .....		23
3.1.4.5	Programming Device .....		24
3.1.4.6	Console .....		24
3.1.4.7	Display Monitoring .....		24
3.1.4.8	Memori .....		25
3.1.4.9	Koneksi Peralatan Input / Output dengan PLC .....		



3.2 Bahasa Pemrograman PLC .....	26
3.2.1 <i>Function Chart</i> .....	27
3.2.1.1 Membuat Program Baru .....	27
3.2.1.2 <i>Area</i> Pengerjaan dari FCH Editor .....	28
3.2.1.3 <i>New Element</i> .....	29
3.2.1.3.1 <i>Network Input</i> .....	29
3.2.1.3.2 <i>Network Output</i> .....	30
3.2.1.3.3 <i>Negation</i> .....	30
3.2.1.3.4 <i>Wire Knot</i> .....	30
3.2.1.3.5 <i>Jump</i> .....	31
3.2.1.3.6 <i>Parameter String</i> .....	31
3.2.1.3.7 <i>Intermediate Variable</i> .....	31
3.2.1.3.8 <i>Boolean Operation</i> .....	32
3.2.1.3.8.1 <i>AND</i> .....	32
3.2.1.3.8.2 <i>OR</i> .....	32
3.2.1.3.8.3 <i>Exclusive OR (ExOr)</i> .....	33
3.2.1.3.9 <i>Flip-flop</i> .....	33
3.2.1.3.9.1 <i>S/R (Reset Dominant)</i> .....	33
3.2.1.3.9.2 <i>R/S (Set Dominant)</i> .....	34
3.2.1.3.10 <i>Timer</i> .....	34
3.2.1.3.10.1 <i>Pulse Timer</i> .....	35
3.2.1.3.10.2 <i>Timer Switch-On Delay</i> .....	35

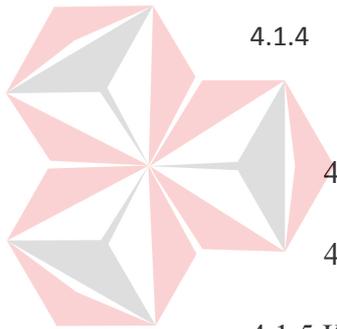




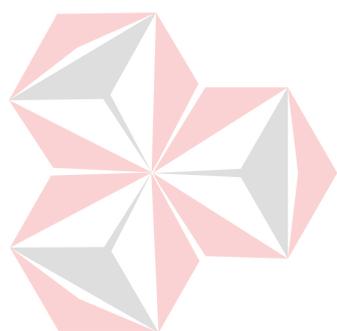
3.2.1.3.10.3 Timer Switch-Off Delay .....	36
3.2.1.3.11 Counter .....	37
3.2.1.3.12 Special Assignments .....	37
3.2.1.3.12.1 One-bit Assignment .....	38
3.2.1.3.12.2 Multibit Assignment .....	38
3.2.1.3.13 Arithmetic Operation .....	38
3.2.1.3.13.1 Addition .....	39
3.2.1.3.13.2 Subtraction .....	39
3.2.1.3.13.3 Division .....	39
3.2.1.3.13.4 Multiplication .....	40
3.2.1.3.14 Comparison Operation .....	40
3.2.1.3.14.1 Greater Than .....	40
3.2.1.3.14.2 Equal or Greater Than .....	41
3.2.1.3.14.3 Equal .....	41
3.2.1.3.14.4 Less Than .....	41
3.2.1.3.14.5 Equal or Less Than .....	42
3.2.1.3.14.6 Not Equal .....	42
3.2.1.4 Draw Wire .....	42
3.2.1.5 Modify/Enter .....	43
3.2.1.6 Shift Element .....	43
3.2.1.7 Delete Element .....	44
3.2.1.8 Line/Coloum .....	44



3.4.3.3 SR1-SR7 .....	57
3.4.3.4 F287-F294 .....	57
<b>BAB IV METODE KERJA PRAKTEK .....</b>	<b>58</b>
4.1 Metode Kerja Praktek .....	58
4.1.1 Pengumpulan Informasi .....	59
4.1.2 Studi Literatur .....	59
4.1.3 Perancangan <i>Hardware</i> .....	59
4.1.4 Perancangan Perangkat Lunak ( <i>Software</i> ) .....	59
4.1.4.1 Perancangan Perangkat Lunak pada PLC .....	60
4.1.4.2 Cara Kerja Program .....	62
4.1.5 Konsultasi Permasalahan .....	65
4.1.6 Perancangan <i>Design</i> .....	65
4.1.7 Pengujian Alat .....	66
<b>BAB V HASIL DAN PEMBAHASAN .....</b>	<b>67</b>
5.1 Mode <i>Sink</i> .....	67
5.2 Mode PLC .....	69
<b>BAB VI PENUTUP .....</b>	<b>70</b>
6.1 Kesimpulan .....	70
6.2 Saran .....	71



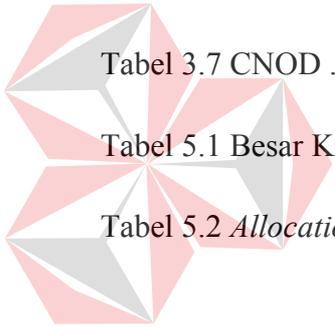
DAFTAR PUSTAKA .....	72
LAMPIRAN .....	74



UNIVERSITAS  
**Dinamika**

## DAFTAR TABEL

	Halaman
Tabel 3.1 Petunjuk Pengisian Pada Program <i>Selection</i> .....	28
Tabel 3.2 Keterangan Simbol <i>Pulse Timer</i> .....	35
Tabel 3.3 Keterangan Simbol Timer <i>Switch-On Delay</i> .....	36
Tabel 3.4 Keterangan Simbol Timer <i>Switch-Off Delay</i> .....	36
Tabel 3.5 Keterangan Simbol <i>Counter</i> .....	37
Tabel 3.6 <i>Start</i> dan <i>End</i> dari <i>Wire</i> .....	43
Tabel 3.7 CNOD .....	55
Tabel 5.1 Besar Kecepatan Motor Tiga Fasa Pada <i>Inverter VF-S11</i> .....	68
Tabel 5.2 <i>Allocation List</i> PLC .....	69



UNIVERSITAS  
Dinamika

## DAFTAR GAMBAR

	Halaman
Gambar 3.1 Sistem Komponen PLC .....	21
Gambar 3.2 Tampilan Program <i>Selection</i> .....	28
Gambar 3.3 <i>Area</i> Pengerjaan dari <i>FCH Editor</i> .....	29
Gambar 3.4 Simbol <i>Network Input</i> .....	30
Gambar 3.5 Simbol <i>Network Output</i> .....	30
Gambar 3.6 a) Simbol <i>Negation Input</i> , b) Simbol <i>Negation Output</i> .....	30
Gambar 3.7 Simbol <i>Wire Knot</i> .....	31
Gambar 3.8 Simbol <i>Jump</i> .....	31
Gambar 3.9 Simbol Parameter String .....	31
Gambar 3.10 Simbol <i>Intermediate Variable</i> .....	31
Gambar 3.11 Simbol <i>AND</i> .....	32
Gambar 3.12 Simbol <i>OR</i> .....	33
Gambar 3.13 Simbol <i>Exlusive OR</i> .....	33
Gambar 3.14 Simbol S/R ( <i>Reset Dominant</i> ) .....	34
Gambar 3.15 Simbol R/S ( <i>Set Dominant</i> ) .....	34
Gambar 3.16 Simbol <i>Pulse Timer</i> .....	35
Gambar 3.17 Simbol Timer <i>Switch-On Delay</i> .....	36
Gambar 3.18 Simbol Timer <i>Switch-Off Delay</i> .....	36

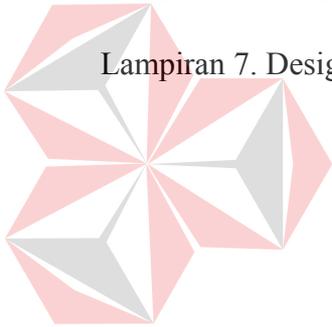
Gambar 3.19 <i>Counter</i> .....	37
Gambar 3.20 Simbol <i>One-bit Assignment</i> .....	38
Gambar 3.21 Simbol <i>Multibit Assignment</i> .....	38
Gambar 3.22 Simbol <i>Addition</i> .....	39
Gambar 3.23 Simbol <i>Subtraction</i> .....	39
Gambar 3.24 Simbol <i>Division</i> .....	40
Gambar 3.25 Simbol <i>Multiplication</i> .....	40
Gambar 3.26 Simbol <i>Greater Than</i> .....	40
Gambar 3.27 Simbol <i>Equal or Greater Than</i> .....	41
Gambar 3.28 Simbol <i>Equal</i> .....	41
Gambar 3.29 Simbol <i>Less Than</i> .....	41
Gambar 3.30 Simbol <i>Equal or Less Than</i> .....	42
Gambar 3.31 Simbol <i>Not Equal</i> .....	42
Gambar 3.32 Tampilan <i>Command Shift Element</i> .....	44
Gambar 3.33 Tampilan <i>Command New Network</i> .....	45
Gambar 3.34 Beda Fasa pada RST .....	46
Gambar 3.35 Struktur Motor Tiga Fasa .....	47
Gambar 3.36 Struktur <i>Star</i> .....	47
Gambar 3.37 Hubungan Struktur <i>Star</i> pada Motor Tiga Fasa .....	48
Gambar 3.38 Struktur Delta .....	49
Gambar 3.39 Hubungan Struktur Delta pada Motor Tiga Fasa .....	49
Gambar 3.40 Rangkaian Pembalik Arah Kekanan .....	49

Gambar 3.41 Rangkaian Pembalik Arah Kekiri .....	50
Gambar 3.42 <i>Terminal Board</i> .....	52
Gambar 3.43 Mode <i>Source</i> .....	52
Gambar 3.44 Mode <i>Sink</i> .....	53
Gambar 3.45 Mode <i>PLC</i> .....	54
Gambar 4.1 Alur Metode Kerja Praktek .....	58
Gambar 4.2 Blok Diagram Perancangan <i>Hardware</i> .....	59
Gambar 4.3 <i>Flowchart</i> Perangkat Lunak pada PLC .....	61
Gambar 4.4 <i>Network 1</i> .....	62
Gambar 4.5 <i>Network 2</i> .....	63
Gambar 4.6 <i>Network 3</i> .....	64
Gambar 4.7 <i>Network 4</i> .....	65



## DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Kartu Bimbingan .....	74
Lampiran 2. Acuan Kerja .....	75
Lampiran 3. Program <i>Function Chart</i> .....	83
Lampiran 4. Fungsi <i>Terminal-terminal</i> Pada <i>Terminal Board</i> .....	84
Lampiran 5. Parameter <i>Inverter</i> .....	86
Lampiran 6. Fungsi dari <i>Input</i> dan <i>Output</i> Pada Fungsi Modul .....	98
Lampiran 7. Design Pengaksesan Motor Tiga Fasa Melalui Inverter VF-S11	99



UNIVERSITAS  
**Dinamika**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

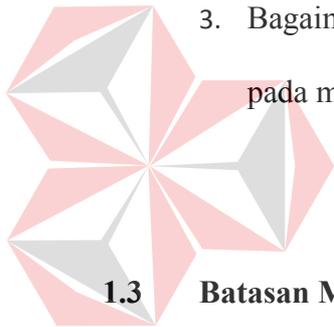
Dengan berkembangnya ilmu pengetahuan dan teknologi pada jaman sekarang ini manusia dituntut eksis dan harus selalu untuk mencari hal-hal yang baru dan dengan hal yang baru itu kita harus mengetahui tentang apa yang dibutuhkan oleh masyarakat saat ini untuk mempermudah pekerjaan mereka. Oleh karena itu untuk meningkatkan mutu pendidikan di Indonesia terutama di STIKOM (Sekolah Tinggi Manajemen Informatika dan Teknik Komputer) Surabaya, mahasiswa STIKOM diberikan suatu mata kuliah kerja praktek untuk melatih mahasiswa merasakan suasana kerja yang penuh tanggung jawab dan tekanan suatu pekerjaan yang diberikan.

Pada saat ini banyak bermunculan pabrik-pabrik atau instansi-instansi yang sebagian sistem alat produksinya menggunakan mikrokontroler dan PLC sehingga sekarang ini banyak di butuhkan tenaga ahli yang professional dalam bidang mikrokontroler dan PLC. Salah satu perguruan tinggi yang dapat menciptakan tenaga ahli yang professional dalam bidang mikrokontroler dan PLC adalah STIKOM dimana di dalamnya terdapat jurusan sistem komputer yang mendalami tentang mikrokontroler dan PLC, maka dari itu kami melaksanakan kerja praktek di Laboratorium PLC STIKOM Surabaya guna untuk menambah wawasan tentang PLC selain yang telah kami dapatkan di mata perkuliahan biasa.

## 1.2 Perumusan Masalah

Berdasarkan permasalahan yang diketahui di atas terdapat rumusan masalah dalam laporan kerja praktek yaitu:

1. Bagaimana mengakses motor tiga fasa menggunakan *inverter* VF-S11 dengan tombol yang ada pada *operation panel* VF-S11.
2. Bagaimana mengontrol motor tiga fasa menggunakan *inverter* VF-S11 pada mode *sink*.
3. Bagaimana mengakses motor tiga fasa menggunakan *inverter* VF-S11 pada mode PLC.



1.3

## Batasan Masalah

UNIVERSITAS  
Dinamika

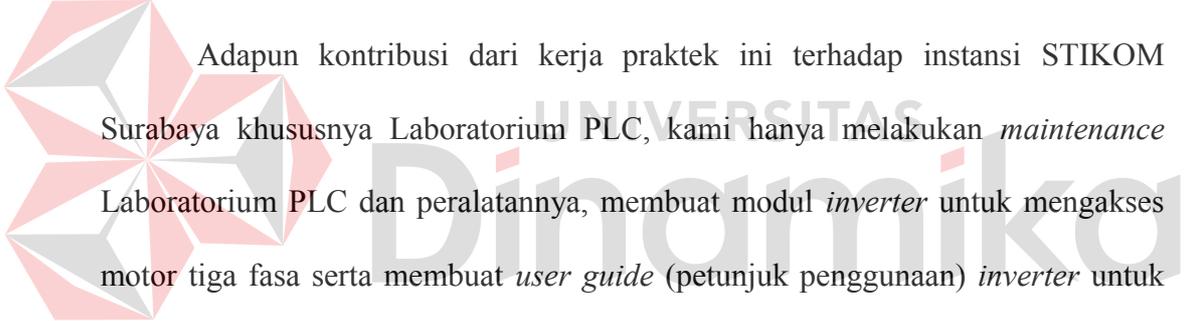
Adapun batasan masalah kerja praktek yaitu:

1. Mempelajari bahasa pemrograman *function chart* pada PLC FESTO di Laboratorium PLC.
2. *Inverter* yang digunakan bertipe VF-S11 buatan Toshiba.
3. Melakukan pengontrolan *inverter* motor 3 fasa dengan menggunakan pemrograman *function chart*.

#### 1.4 Tujuan

Tujuan dari kerja praktek ini adalah supaya mahasiswa belajar membiasakan diri menghadapi lingkungan kerja yang sebenarnya yang seringkali berbeda antara teori, praktek dengan kenyataan di lapangan. Serta mahasiswa akan memiliki suatu pengalaman tentang dunia kerja. Mampu menerapkan ilmu yang didapat selama kuliah untuk menganalisa dan permasalahan dalam sistem yang lebih luas.

#### 1.5 Kontribusi



Adapun kontribusi dari kerja praktek ini terhadap instansi STIKOM Surabaya khususnya Laboratorium PLC, kami hanya melakukan *maintenance* Laboratorium PLC dan peralatannya, membuat modul *inverter* untuk mengakses motor tiga fasa serta membuat *user guide* (petunjuk penggunaan) *inverter* untuk mengakses motor tiga fasa yang dikontrol melalui PLC.

#### 1.6 Sistematika Penulisan

Penulisan laporan disusun dengan sistematika sebagai berikut :

### **BAB I PENDAHULUAN**

Bab ini membahas tentang uraian mengenai latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan, kontribusi, serta sistematika penulisan.

## **BAB II GAMBARAN UMUM STIKOM SURABAYA**

Bab ini berisi sejarah dan perkembangan, lokasi, jenis usaha, visi, misi dan komitmen STIKOM Surabaya sebagai tempat kerja praktek.

## **BAB III LANDASAN TEORI**

Bab ini membahas tentang teori penunjang yang digunakan sebagai acuan dalam kerja praktek tersebut.

## **BAB IV METODE KERJA PRAKTEK**

Bab ini membahas tentang metode yang digunakan dalam pengerjaan kerja praktek dari tahap awal sampai penulisan laporan.

## **BAB V HASIL DAN PEMBAHASAN**

Bab ini membahas tentang proses pembuatan program serta menampilkan foto-foto alat yang kita kerjakan.

## **BAB VI PENUTUP**

Bab ini adalah bagian terakhir dari laporan kerja praktek yang membahas tentang kesimpulan dari keseluruhan hasil kerja praktek serta saran disesuaikan dengan hasil dan pembahasan pada bab-bab yang sebelumnya.

## BAB II

### GAMBARAN UMUM STIKOM SURABAYA

#### 2.1 Sejarah dan perkembangan

Dalam perkembangannya, teknologi informasi telah mengambil alih tugas-tugas operasional yang rutin, rumit dan repetitive, dan dengan bantuan komputer, membuat tugas-tugas tersebut menjadi bersifat otomatis.

Agar kehadiran teknologi canggih itu dapat dimanfaatkan secara maksimal dalam rangka meningkatkan pembangunan nasional maka perlu tersedia sumber daya manusia yang terdidik, berkualitas, dan memiliki keterampilan dalam penguasaan teknologi informatika.

Untuk menyediakan sumber daya manusia yang terdidik dan berkualitas serta memiliki penguasaan dalam teknologi informatika maka STIKOM Surabaya hadir untuk memenuhi kebutuhan tersebut. STIKOM Surabaya adalah sekolah tinggi khusus mengeluarkan sarjana-sarjana yang benar-benar tahu akan perkembangan teknologi informasi.

Menyadari hal ini, tanggal 30 April 1983 untuk pertama kali di wilayah JATIM (Kopertis Wilayah VII) dibuka pendidikan tinggi komputer, Akademi Komputer & Informatika Surabaya (AKIS) berdasarkan SK No. 01/KPT/PB/III/1983. Tokoh pendirinya saat itu adalah :

- Laksada TNI (Purn) Mardiono

- Ir. Andrian A. T.
- Ir. Handoko Anindyo
- Dra. Suzanna Surojo
- Dra. Rosy Merianti, AK.

### **2-3 Maret 1984**

Kepanjangan AKIS dirubah menjadi Akademi Manajemen Informatika dan Komputer Surabaya yang bertempat di Jl. Ketintang Baru XIV/2

### **10 Maret**

Memperoleh ijin Operasional penyelenggaraan program Diploma III Manajemen Informatika dengan keputusan nomor : 061/Q/1984 dari Direktorat Jendral Pendidikan Tinggi (Dikti) melalui koordinator Kopertis wilayah VII.

### **19 Juni 1984**

AKIS memperoleh status TERDAFTAR berdasarkan keputusan Dikti nomor : 0274/O/1984 dan kepanjangan AKIS berubah menjadi Akademi Manajemen Informatika dan Teknik Komputer Surabaya. Berdasarkan SK Dirjen DIKTI nomor : 45/DIKTI/KEP/1992, status DIII Manajemen Informatika ditingkatkan menjadi DIAKUI.

### **30 Maret 1986**

AKIS ditingkatkan menjadi Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya (STIKOM Surabaya).

**25 November 1986**

STIKOM mendapatkan status TERDAFTAR untuk program S1 dan DIII Manajemen Informatika.

**11 Desember 1987**

Peresmian gedung STIKOM Surabaya Jl. Kutisari 66 Surabaya oleh Bapak Wahono Gubernur Jawa Timur saat itu. Membuka bidang studi D1 Program Studi Komputer Akutansi.

**1990**

Membuka bidang studi D1 Program Studi Komputer Keuangan / Perbankan.

**1 Januari 1992**

Membuka Program S1 Jurusan Teknik Komputer dengan status TERDAFTAR.

**19 Maret 1992**

DII Manajemen Informatika memperoleh status DIAKUI.

**21 Januari 1993**

Program S1 Manajemen Informatika memperoleh status DIAKUI.

**1 November 1994**

Membuka program D1 Program Studi Komputer Grafik multimedia.



UNIVERSITAS  
Dinamika

### **31 Januari 1995**

STIKOM Surabaya memperoleh kenaikan status DISAMAKAN untuk program S1 dan DIII Manajemen Informatika.

### **September 1997**

Merupakan awal pembangunan gedung kampus II STIKOM Surabaya daerah Kedung Baruk, tepatnya Jl. Raya Kedung Baruk 98, yang terdiri 3 gedung perkuliahan dengan masing-masing berlantai 9, 1 gedung rektorat berlantai 1, beserta sarana dan prasarana yang lengkap & memadai.

### **28 Oktober 1997**

Pemancangan tiang pertama gedung baru STIKOM Surabaya di Jl. Raya Kedung Baruk 98 Surabaya.

### **12 Mei 1998**

STIKOM membuka tiga program pendidikan baru sekaligus, yaitu :

1. DIII bidang studi Sekretari Berbasis Komputer.
2. DII bidang studi Komputer Grafik Multimedia.
3. DI bidang studi Jaringan Komputer.

### **30 Juni 1998**

STIKOM menerima status DISAMAKAN untuk :

1. Program DII bidang studi Komputer Grafik Multimedia.

2. Program D1 semua bidang studi (Komputer Akuntansi, Jaringan Komputer, Komputer Grafik Multimedia).

### **Desember 1998**

Memperoleh status TERAKREDITASI dari Badan Akreditasi Nasional, untuk Program Pendidikan S1 Jurusan Manajemen Informatika (SK BAN-PT No. 002/BAN-PT/Ak-II/XII/1998 DENGAN NILAI 542,2(B)).

### **Maret 1999**

Perpindahan kampus baru, Jl. Raya Kedung Baruk 98, dari kampus STIKOM SIER Jl. Rungkut Industri I / 1 Surabaya.

### **Juni 1999**

Pemisahan program studi DI Grafik Multimedia menjadi program studi.

DI Grafik dan program studi DI Multimedia menjadi program studi DII Multimedia.

### **Agustus 2000**

Memperoleh status TERAKREDITAS dari badan Akreditasi Nasional, untuk Program Pendidikan S1 Jurusan Teknik Komputer (SK BAN-PT No. 109/BAN-PT/Ak-IV/VIII/2000 DENGAN NILAI 535 (B)).

**Mei 2002**

Memperoleh status TERAKREDITASI dari bahan Akreditasi Nasional, untuk Program Pendidikan DIII Jurusan Manajemen Informatika (SK BAN-PT No. 002/BAN-PT/Ak-I/Dpl-III/2002 DENGAN NILAI 332 (B)).

**Awal tahun 2003**

Pembangunan tower 1 selesai.

**Juli 2003**

Membuka bidang studi DIII Program Studi Komputer Percetakan & Kemasan.

**13 Agustus 2003**

Program Studi Strata 1 Teknik Komputer berubah nama menjadi Program Studi Strata 1 Sistem Komputer.

**2 September 2003**

Program Studi Srata 1 Manajemen Informatika berubah nama menjadi Program Studi Srata 1 Sistem Informasi.

**Tahun 2003**

Program Studi Diploma III Sekretaris Berbasis Komputer berubah nama menjadi Komputer Sekretaris & Perkantoran Modern. Program Studi Diploma 1 dan II di non aktifkan.

**4 Juni 2004**

STIKOM Surabaya memperoleh status TERAKREDITASI (B) untuk Jurnal Ilmiah GEMATIKA (Jurnal Manajemen Informatika) dan TERAKREDITASI (C) untuk Jurnal Ilmiah GEMATEK (Jurnal Teknik Komputer).

**24 Juli 2004**

Dilakukan tiang pancang pertama untuk tower II.

**3 Maret 2005**

Program Studi Diploma III Komputer Akutansi memperoleh ijin perpanjangan berdasarkan surat dari Dikti tentang Ijin Penyelenggaraan nomor : 644/D/T/2005.

**10 Mei 2005**

Program Studi DIII Komputer Multimedia memperoleh ijin perpanjangan berdasarkan surat dari Dikti nomor : 1402/D/T/2005.

**3 Juni 2005**

Program Studi Strata 1 Sistem Informasi memperoleh status TERAKREDITASI berdasarkan SK BAN-PT Nomor : 012/BAN-PT/Ak-IX/S1/VII/2005 dengan nilai 333 (B)).

**27 Desember 2005**

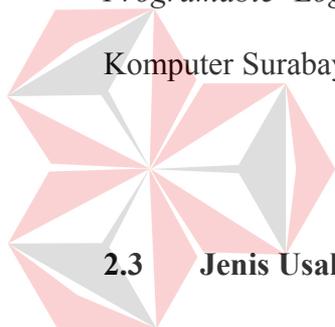
Program Studi Strata 1 Sistem Komputer memperoleh status TERAKREDITASI berdasarkan SK BAN-PT Nomor : 024/BAN-PT/Ak-IX/S1/XII/2005 dengan nilai 330 (B)).

**16 Januari 2005**

Program Studi Diploma III Komputer Sektetaris & Perkantoran Modern berubah nama menjadi Program Diploma III Komputerisasi Perkantoran dan Kesekretariatan berdasar surat dari Dikti tentang Ijin Penyelenggaraan nomor : 75/D/T/2006.

## **2.2 Lokasi**

Lokasi kerja praktek diselenggarakan di Laboratorium (Lab.) *Programable Logic Control* (PLC) Sekolah Tinggi Manajemen dan Teknik Komputer Surabaya yang terletak di Jl. Raya Kedung Baruk 98 Surabaya lantai 8.



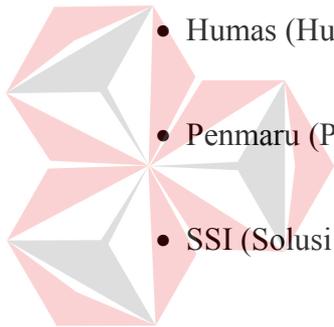
## **2.3 Jenis Usaha**

Adapun kegiatan usaha STIKOM Surabaya adalah menyelenggarakan jasa pendidikan Perguruan Tinggi. Dalam operasionalnya STIKOM Surabaya dibagi menjadi beberapa departemen, dimana dari masing-masing departemen akan bertanggungjawab terhadap spesifikasi pekerjaan tertentu. Departemen tersebut meliputi :

- PSDM (Pengembangan Sumber Daya Manusia)
- BAAK (Bagian Administrasi dan Akademik)
- AU (Administrasi Umum)

UNIVERSITAS  
**Dinamika**

- Prodi (Program Studi)
- Labkom (Laboratorium Komputer)
- PA (Penelitian Akademik)
- PM (Pengabdian Masyarakat)
- PPTI (Pengembangan dan Penerapan Teknologi Informasi)
- Perpustakaan
- Kemahasiswaan
- Humas (Hubungan Masyarakat)
- Penmaru (Penerimaan Mahasiswa Baru)
- SSI (Solusi Sistem Informasi)
- PPKF (Pusat Pelatihan Kemasan Fleksibel)
- Keuangan
- BD (Business Development)
- Research Center
- SPROM (STIKOM Promotion Movement)
- Sekretaris Eksekutif Ketua
- Kendali Mutu



UNIVERSITAS  
Dinamika

## 2.4 Visi, Misi dan Komitmen STIKOM Surabaya

### 2.4.1 Visi STIKOM Surabaya

Tercapainya kepeloporan karena keunggulan manusia pada peringkat benchmark yang pada tahun 2018 mendekati keunggulan sumber daya manusia Singapura dalam upaya mendukung keunggulan studi dalam arti luas tentang Teknologi Informasi (TI) untuk menjamin kesejahteraan manusia yang pluralisme dan multikulturalisme.

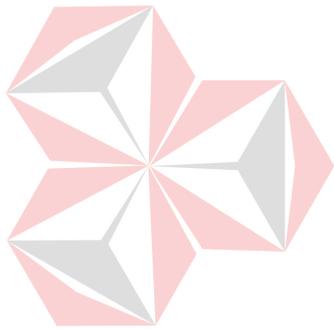
### 2.4.2 Misi STIKOM Surabaya

1. Meningkatkan strata Pendidikan secara terus-menerus untuk semua lapisan pada bauran kompetensi, bukan hanya pada kognisi dan pada psikomotor keahlian, tapi terutama pada kompetensi sikap mental untuk semua manusianya yang ada di STIKOM Surabaya agar semakin produktif dan inspiratif, dengan hidup hanya melayani dan melayani.
2. Mengembangkan corporate governance yang sehat dan produktif secara sistematis tapi bersifat emerging, demi terciptanya habitat organisasi yang *socio-cultural economic* sekaligus inovatif.
3. Melakukan integrasi mulai dari perolehan intake mahasiswa walaupun pada standar biasa, tetapi akan selalu dijaga dan diproses tidak hanya pada tingkat maksimum tetapi terutama optimum, sampai dengan suatu hasil outcome yang luar biasa pada *hardskill* dan terutama pada *softkill*-nya, demi perkembangan masyarakat, negara dan bangsa.

4. Meningkatkan produktivitas dengan mengoptimalkan pengelolaan sumber daya, terutama sekali sumber daya manusianya dan sumber daya keuangan berdasarkan pada kegiatan yang relevan dan sesuai dengan harkat manusia.
5. Meningkatkan kesejahteraan untuk semua manusianya berdasarkan keseimbangan pada keadilan dan prestasi kontribusi organisasional setiap anggota organisasi di STIKOM Surabaya ini secara merata dan menyeluruh.
6. Melakukan perluasan pengabdian masyarakat, berbasis pada pengembangan ilmu dan teknologi yang dikuasai, untuk peningkatan kesejahteraan semua manusia, khususnya dengan peduli pada kaum miskin.
7. Melakukan peningkatan dan penajaman serta perluasan semangat penelitian, bukan demi ilmu dan teknologi itu sendiri, tapi untuk mengungkap kebenaran realitas kehidupan agar kehidupan manusianya lebih manusiawi dan manusianya berguna bagi seluruh umat manusia.
8. Berjejaring secara proaktif dan sehat dengan *stakeholders* untuk konsolidasi dan adaptasi organisasi dalam rangka perkembangan dan pertumbuhan organisasi.

### 2.4.3 Komitmen STIKOM Surabaya

Dengan saling bergandengan tangan baik kedalam maupun keluar, semoga visi dan misi ini tidak hanya dipahami, tetapi juga dihayati dan lebih daripada itu dilaksanakan secara konsisten dan semakin meningkat oleh setiap individu yang berada di STIKOM Surabaya untuk menghidupi (bukan mencari kehidupan dari) STIKOM Surabaya yang kita banggakan dan akan terus kita cintai dengan semangat hanya melayani dan melayani.



UNIVERSITAS  
**Dinamika**

## BAB III

### LANDASAN TEORI

#### 3.1 PLC

PLC adalah Suatu alat yang digunakan untuk mengontrol dengan menggunakan logika dan bisa diprogram sesuai kebutuhan.

- *Programable* : Dapat diprogram sesuai kebutuhan.
- *Logic* : Terdapat fungsi-fungsi logika seperti and, or, not, penjumlahan, pembagian, perkalian, dan sebagainya.
- *Controller* ; sebagai alat control.

Dalam bidang industri penggunaan mesin otomatis dan pemrosesan secara otomatis merupakan hal yang umum. Sistem pengontrolan dengan elektromekanik yang menggunakan relay-relay mempunyai banyak kelemahan, diantaranya kontak-kontak yang dipakai mudah hangus karena panas atau terbakar atau karena hubung singkat, membutuhkan biaya yang besar saat instalasi, pemeliharaan dan modifikasi dari sistem yang telah dibuat jika dikemudian hari diperlukan modifikasi.

Dengan menggunakan PLC hal-hal ini dapat diatasi, karena sistem PLC mengintegrasikan berbagai macam komponen yang berdiri sendiri menjadi suatu sistem kendali terpadu dan dengan mudah merenovasi tanpa harus mengganti semua instrumen yang ada.

Sumber : Himpunan Mahasiswa Elektro Universitas Satyagama. 2001. *Pengenalan Programmable Logic Controller (PLC)* (Online). [http://www.migas-indonesia.com/files/article/PENGENALAN\\_PLC.doc](http://www.migas-indonesia.com/files/article/PENGENALAN_PLC.doc), diakses tanggal 19 Januari 2010.

### 3.1.1 Sejarah

Pada awalnya PLC digunakan sebagai alat elektronik untuk pengganti panel relay. Pada saat itu PLC hanya bekerja untuk kondisi ON-OFF untuk pengendalian motor, solenoid, dan actuator. Alat ini mampu mengambil keputusan yang lebih baik dibandingkan relay biasa. PLC pertama kali banyak digunakan pada bagian otomotif. Sebelum adanya PLC, sudah banyak peralatan *control sequence*. Ketika relay muncul, panel control dengan relay menjadi *control sequence* yang utama.

Pada tahun 1978, penemuan chip mikroprosesor menaikkan kemampuan komputer untuk segala jenis sistem otomatisasi dengan harga yang terjangkau. Robotika, peralatan otomatis dan komputer dari berbagai tipe termasuk PLC berkembang dengan pesat. Program PLC makin mudah untuk dimengerti oleh banyak orang.

Pada awal tahun 1980 PLC makin banyak digunakan. Beberapa perusahaan elektronik dan komputer membuat PLC dalam volume yang besar, meskipun peralatan industri mesin CNC telah digunakan beberapa waktu yang lalu, PLC tetap digunakan. PLC juga digunakan untuk sistem otomatisasi building dan juga *security control system*.

Sekarang sistem kontrol sudah meluas hingga keseluruhan pabrik dan sistem kontrol total dikombinasikan dengan kontrol *feedback*, pemrosesan data, dan sistem monitor terpusat. Saat ini PLC sudah menjadi alat yang cerdas, yang merupakan kebutuhan utama di industri modern.

Sumber : Hidayat, Andhika. 2010. *Dasar-dasar PLC* (Online). <http://www.scribd.com/doc/25696-4898/Dasar-Dasar-PLC>, diakses tanggal 19 Januari 2010.

### 3.1.2 Konsep PLC

Konsep dari PLC sesuai dengan namanya adalah sebagai berikut :

*Programmable* : menunjukkan kemampuannya yang dapat dengan mudah diubah-ubah sesuai program yang dibuat dan kemampuannya dalam hal memori program yang telah dibuat.

*Logic* : menunjukkan kemampuannya dalam memproses input secara aritmetik (ALU), yaitu melakukan operasi membandingkan, menjumlahkan, mengalikan, membagi, mengurangi dan negasi.

*Controller* : menunjukkan kemampuannya dalam mengontrol dan mengatur proses sehingga menghasilkan output yang diinginkan.

Sumber : Himpunan Mahasiswa Elektro Universitas Satyagama. 2001. *Pengenalan Programmable Logic Controller (PLC)* (Online). [http://www.migas-indonesia.com/files/article/PENGENALAN\\_PLC.doc](http://www.migas-indonesia.com/files/article/PENGENALAN_PLC.doc), diakses tanggal 19 Januari 2010.

### 3.1.3 Fungsi PLC

Fungsi dan kegunaan dari PLC dapat dikatakan hampir tidak terbatas. Tapi dalam prakteknya dapat dibagi secara umum dan khusus.

Secara umum fungsi dari PLC adalah sebagai berikut :

1. Kontrol Sekuensial

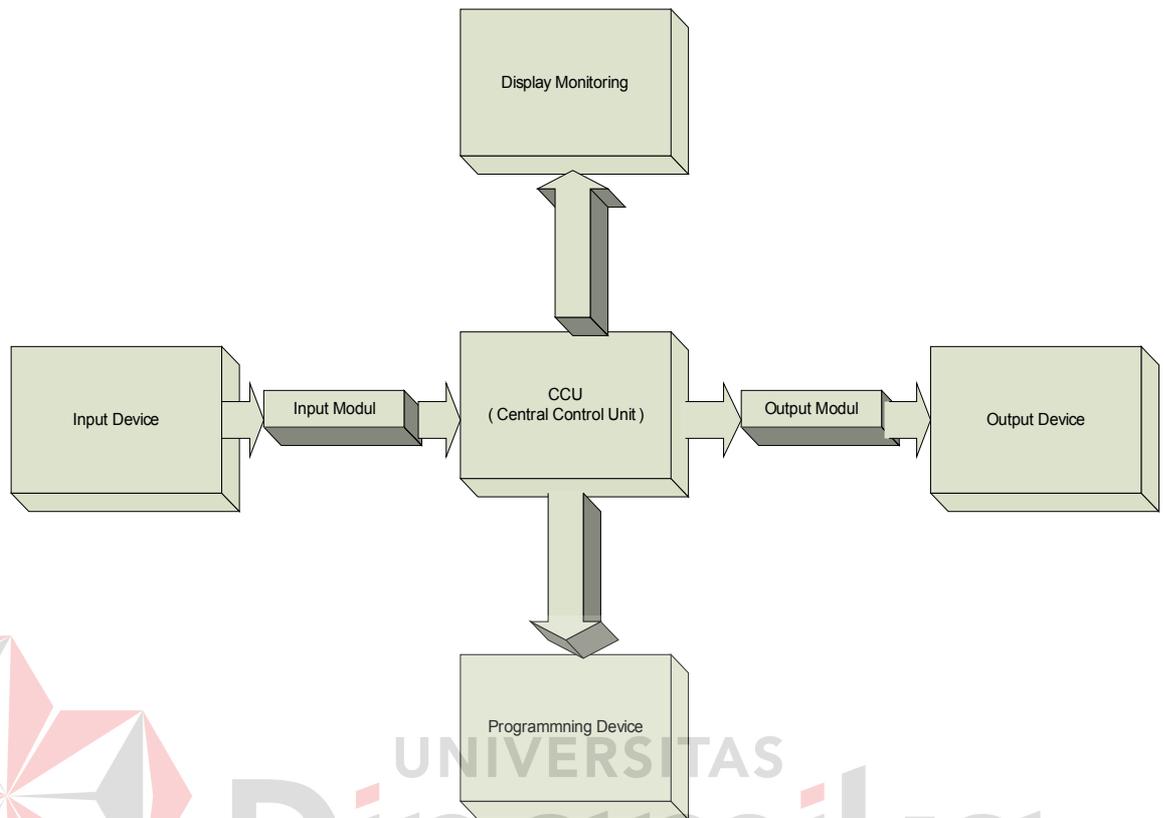
PLC memproses *input* sinyal biner menjadi *output* yang digunakan untuk keperluan pemrosesan teknik secara berurutan (sekuensial), disini PLC menjaga agar semua *step* / langkah dalam proses sekuensial berlangsung dalam urutan yang tepat.

2. Monitoring Plant

PLC secara terus menerus memonitor suatu sistem (misalnya temperatur, tekanan, tingkat ketinggian) dan mengambil tindakan yang diperlukan sehubungan dengan proses yang dikontrol (misalnya nilai sudah melebihi batas) atau menampilkan pesan tersebut ke operator.

Sumber : Himpunan Mahasiswa Elektro Universitas Satyagama. 2001. *Pengenalan Programmable Logic Controller (PLC)* (Online). [http://www.migas-indonesia.com/files/article/PENGENALAN\\_PLC.doc](http://www.migas-indonesia.com/files/article/PENGENALAN_PLC.doc), diakses tanggal 19 Januari 2010.

### 3.1.4 Sistem Komponen PLC



Gambar 3.1 Sistem Komponen PLC

#### 3.1.4.1 CCU ( *Central Control Unit* )

CCU merupakan otak dari PLC, dimana ia akan mengendalikan dan mengawasi jalannya operasi dalam PLC sesuai dengan instruksi program yang tersimpan dalam memori. Suatu jalur komunikasi internal akan membawa informasi dari dan ke CCU, memori, unit I/O, dengan dikendalikan oleh CCU. Sistem CCU pada PLC berbasis mikroprosesor. Adapun jenis *microprocessor* yang dipergunakan tergantung dari merk dan tipe PLC-nya. CCU atau disebut juga CPU (*Central Processing Unit*), yang terdiri atas bagian : *processor, memory, dan power supply*.

### 3.1.4.2 Modul Input / Output

Modul I/O dari suatu PLC merupakan komunikasi atau hubungan PLC dengan dunia luar. Dengan modul ini maka PLC mampu mengendalikan suatu proses. Unit I/O ini mempunyai karakteristik yang berbeda-beda sesuai dengan jenis PLC nya. Ada PLC yang mempunyai 8 bit I/O digital tetapi dapat juga terdapat modul extended I/O yang memungkinkan PLC memiliki banyak I/O. Tiap I/O ini mempunyai alamat tersendiri yang akan digunakan pada program. Modul input PLC berhubungan dengan elemen sensor yang memberikan informasi keadaan proses. Sinyal informasi ini akan diolah sesuai dengan program yang telah dibuat oleh CCU. Sedangkan modul output PLC berhubungan dengan elemen aktuator yang akan memberikan aksi kendali kepada plant. Apabila input berupa sinyal analog, maka dibutuhkan suatu modul input analog yang berfungsi sebagai ADC. Modul ini juga akan mengkondisikan sinyal input sehingga range input analog menjadi sesuai dengan range input ADC (*scaling*). Begitu juga dengan output, apabila aktuator membutuhkan sinyal analog maka dibutuhkan modul output analog. Dengan demikian PLC mempunyai kemampuan lebih dengan menerima input dan memberikan output analog dengan pemrosesan sinyal secara digital.

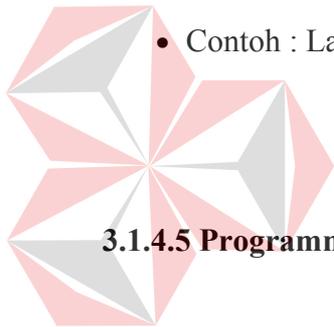
Sumber : Dodi Setyobudi, Aloysius. 1998. *Laporan Kerja Praktek di Politeknik Manufaktur* (Online). [http://www.reocities.com/al\\_dodi/kerja/KP5.pdf](http://www.reocities.com/al_dodi/kerja/KP5.pdf), diakses tanggal 19 Januari 2010.

### 3.1.4.3 *Input Device*

- *Input device* adalah segala sesuatu yang digunakan sebagai *input* untuk PLC.
- Contoh : Tombol, sensor induktif, sensor kapasitif, sensor magnetik, sensor optik dan lain sebagainya.

### 3.1.4.4 *Actuator (Output Device)*

- Segala sesuatu yang menjadi *output* PLC.
- Contoh : Lampu, solenoid valve, motor listrik, buzzer dan lain sebagainya.



### 3.1.4.5 *Programming Device*

Bagian ini merupakan elemen yang berinteraksi dengan pemakai. Alat ini memudahkan pemakai dalam memprogram ataupun mengubah program PLC. Apabila PLC sudah terprogram, maka alat ini tidak diperlukan lagi dan PLC bekerja secara mandiri. Alat ini dapat berupa *handheld programmer/console* berbentuk seperti kalkulator kecil untuk memasukkan program. Programming device dapat juga berupa personal komputer dengan software tertentu yang dikeluarkan oleh pembuat PLC. Masing-masing alat, mempunyai kelebihan dan kekurangan. *Hand-held programmer bentuknya* kecil dan praktis digunakan di lapangan, tetapi tidak komunikatif dengan *user* karena tampilan programnya hanya satu baris. Sedangkan PC tidak mudah dibawa atau dipindahkan ke

lapangan tetapi cara pemrogramannya lebih mudah karena software-nya telah dirancang untuk memudahkan.

Sumber : Dodi Setyobudi, Aloysius. 1998. *Laporan Kerja Praktek di Politeknik Manufaktur* (Online). [http://www.reocities.com/al\\_dodi/kerja/KP5.pdf](http://www.reocities.com/al_dodi/kerja/KP5.pdf), diakses tanggal 19 Januari 2010.

#### **3.1.4.6 Console**

Sebuah perangkat kecil (*portable*) yang berfungsi memasukan instruksi program ke dalam PLC (Benyamin,2009 : 16).

#### **3.1.4.7 Display Monitoring**

Peralatan yang digunakan untuk proses ini sama dengan dengan peralatan programming device hanya difungsikan untuk hal yang berbeda (Benyamin, 2009 : 16).

#### **3.1.4.8 Memori**

Karakteristik terpenting PLC adalah kemudahan pemakai dalam mengganti program dengan mudah dan cepat. Kemudahan ini didapatkan karena arsitektur PLC yang dilengkapi dengan sistem memori. Sistem memori yang dimaksud adalah tempat pada CCU yang dapat menyimpan data-data urutan instruksi ataupun program yang nantinya akan dieksekusi oleh prosesor. Sistem memori PLC terdiri dari dua macam :

- 1) *Executive memori* : atau disebut juga memori sistem operasi. Sistem memori ini adalah tempat menyimpan program yang menangani operasi PLC. Program permanen ini menjalankan aktivitas seluruh sistem seperti eksekusi program, komunikasi peralatan, dan lain-lain. Bagian ini menyimpan instruksi-instruksi *software* seperti instruksi internal relay, block transfer, instruksi aritmatik, dan lain-lain.
- 2) *Application memori* : Sistem ini untuk menyimpan instruksi program yang dimasukkan oleh pemakai untuk menjalankan proses kendali tertentu. Di samping itu terdapat memori penyimpanan status (status register) *input/output* dan status fungsi dalam PLC itu sendiri, seperti timer dan counter.

Sumber : Dodi Setyobudi, Aloysius. 1998. *Laporan Kerja Praktek di Politeknik Manufaktur* (Online). [http://www.reocities.com/al\\_dodi/kerja/KP5.pdf](http://www.reocities.com/al_dodi/kerja/KP5.pdf), diakses tanggal 19 Januari 2010.

### 3.1.4.9 Koneksi Peralatan *Input / Output* dengan PLC

#### 1. Peralatan Input

##### a. Switch

Terdapat 2 jenis switch yaitu :

1. Push button
2. Detent button

##### b. Sensor

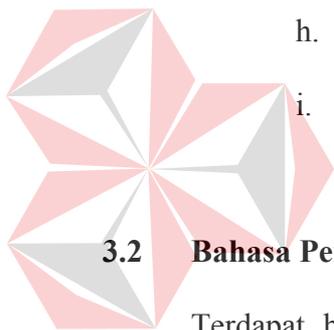
Terdapat 4 jenis sensor yaitu :

1. Sensor warna (Optik)
2. Sensor benda (Capasitor)

3. Sensor magnetic
4. Sensor logam (Induktif)

## 2. Peralatan Output

- a. Relay
- b. Lampu
- c. Buzzer
- d. Transistor
- e. Solenoid valve
- f. Motor listrik
- g. Inverter
- h. Triac
- i. Dan lain sebagainya



### 3.2 Bahasa Pemrograman PLC

Terdapat banyak pilihan bahasa untuk membuat program dalam PLC.

Masing-masing bahasa mempunyai keuntungan dan kerugian tergantung dari sudut pandang kita sebagai *user* / pemogram. Pada umumnya terdapat 4 bahasa dari PLC , yaitu pemrograman *diagram ladder* , *function chart*, *statement list*, dan matrik. Tiap jenis PLC mempunyai karakteristik pemrograman tersendiri. Spesifikasi PLC yang berbeda mempunyai bahasa pemrograman yang berbeda pula. *Diagram Ladder* adalah bahasa yang dimiliki oleh setiap PLC. Dalam menyelesaikan tugas kerja praktek ini, kami menggunakan program *function chart* sebagai pengontrol PLC ke motor tiga fasa melalui *inverter*.

### 3.2.1 *Function Chart*

Program *function chart* dapat digunakan untuk *Festo controller* dari FPC 100 series, FPC 202C dan FPC 404. Bentuk *function chart* sama seperti rangkaian digital sehingga program ini dapat mempermudah *engineer* yang *familiar* dengan elektronika digital dan juga sangat mudah untuk kontrol kombinasional. Prinsip kerjanya sama seperti *ladder diagram*. *Function chart* memiliki persamaan dengan *ladder diagram* yaitu merupakan representasi grafik. *Function chart* memiliki beberapa menu yaitu *FCH editor*, *FCH online display*, *syntax test*, *load program*, *error list*, *FPC online mode*, dan lain sebagainya.

- *FCH editor* digunakan untuk *area* pengerjaan program.
- *FCH online display* digunakan untuk mengecek jalannya program pada saat sedang diload.
- *Syntax test* digunakan untuk mengecek apakah program ada kesalahan atau tidak.
- *Load program* digunakan untuk mengeload program ke PLC.
- *Error list* digunakan untuk melihat daftar *error* jika program ada error.
- *FPC online mode* digunakan untuk mengecek apakah komputer dan PLC sudah terhubung apa belum.

#### 3.2.1.1 Membuat Program Baru

Untuk membuat program baru dengan *function chart* dengan memilih menu *FCH editor*. Setelah memilih *FUC editor* akan keluar menu seperti pada Gambar 3.2.



Gambar 3.2 Tampilan Program Selection.

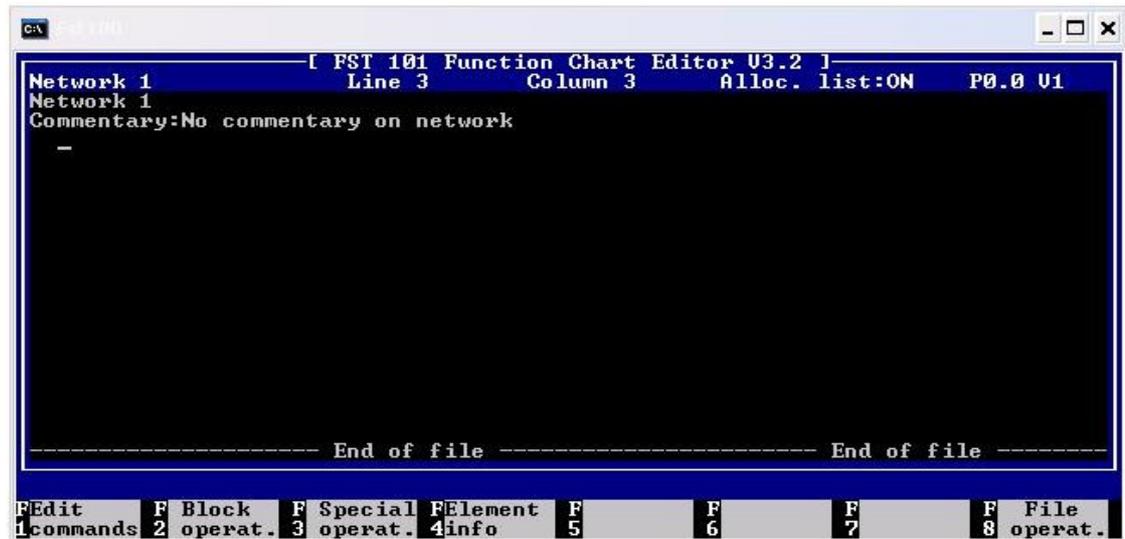
Ketika semua telah dimasukkan, lalu menekan F1. Petunjuk untuk mengisi menu pada program selection dapat dilihat pada Tabel 3.1.

Entry	Keterangan
Prog./Module (P/B)	Pilih P untuk Program atau pilih B untuk Module
Program No.	Untuk pemberian nomer pada program atau module
Version No.	sebagai indikasi
Description	Memberikan komentar atau nama pada program atau <i>module</i>

Tabel 3.1 Petunjuk Pengisian Pada Program Selection.

### 3.2.1.2 Area Pengerjaan dari FCH Editor

Area ini digunakan untuk mengedit atau membuat program, seperti pada Gambar 3.3.



Gambar 3.3 Area Pengerjaan dari FCH Editor.

Untuk mengedit command-command pada program dengan cara tekan F1 (*edit commands*). Didalam menu *edit commands* terdapat beberapa perintah, yaitu : *new elemen* (F1), *draw wire* (F2), *modify/enter* (F3), *shift elemen* (F4), *delete elemen* (F5), *line/coloum* (F6), *new network* (F7), dan *exit* (F8).

### 3.2.1.3 New Element

*New element* digunakan untuk menambahkan elemen pada network.

Didalam *new element* terdapat beberapa menu, diantaranya :

#### 3.2.1.3.1 Network Input

Digunakan untuk menambahkan elemen inputan. Sambungan titik berada disisi kanan simbol. *Network Input* dihubungkan dengan fungsi modul *input*, *intermediate* atau langsung ke *network output* (Bliesener, tanpa tahun : A-2). Cara menambahkan *network input* yaitu tekan F1 (*edit command*) → F1 (*new elemen*) → pilih *network input* → enter.

Gambar 3.4 Simbol *Network Input*.

### 3.2.1.3.2 *Network Output*

Digunakan untuk menambahkan elemen outputan. Sambungan titik berada disisi kanan simbol. *Network output* dihubungkan dengan fungsi modul *input*, *intermediate* atau langsung ke *network input* (Bliesener, tanpa tahun : A-2). Cara menambahkan *network output* yaitu tekan F1 (*edit command*) → F1 (*new elemen*) → pilih *network output* → enter.

Gambar 3.5 Simbol *Network Output*.

### 3.2.1.3.3 *Negation*

Elemen *negation* dapat digunakan pada input ataupun output fungsi modul. Jika inputan dari negation *high* maka outputannya menjadi *low*, begitu juga sebaliknya.

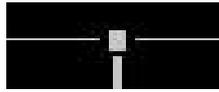
a

b

Gambar 3.6 a) Simbol *Negation Input*, b) Simbol *Negation Output*.

### 3.2.1.3.4 *Wire Knot*

Digunakan untuk menghubungkan antara elemen satu dengan elemen lainnya jika ada percabangan.

Gambar 3.7 Simbol *Wire Knot*.

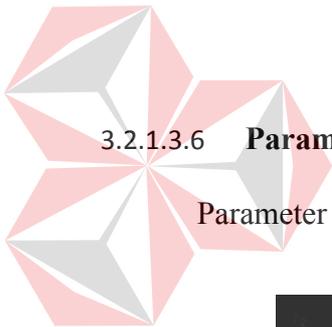
### 3.2.1.3.5 *Jump*

Perintah *jump* ditandai dengan “>>”. Setelah tanda “>>” diikuti dengan nama *network* tujuan. Perintah *jump* dieksekusi saat mendapat sinyal positif (Bliesener, tanpa tahun : A-4).

Gambar 3.8 Simbol *Jump*.

### 3.2.1.3.6 *Parameter String*

Parameter string digunakan untuk inputan string pada program atau modul.



String.....-

Gambar 3.9 Simbol Parameter String.

### 3.2.1.3.7 *Intermediate Variable*

Jika ingin menggunakan elemen untuk *multiple association*, maka harus menggunakan *intermediate variable* untuk menghubungkan antara outputan dengan inputan elemen dari fungsi blok.

-Inter....-

Gambar 3.10 Simbol *Intermediate Variable*.

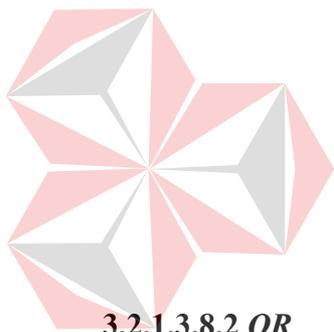
### 3.2.1.3.8 Boolean Operation

Operasi boolean mempunyai 3 elemen yang dapat digunakan, yaitu :

#### 3.2.1.3.8.1 AND

Elemen *AND* ditunjukkan dengan persegi panjang, dengan fungsi pengidentifikasi “&”. Elemen ini mempunyai 2-16 inputan. Sinyal *output* adalah 1, jika semua sinyal *input* adalah 1. Sebuah sinyal *output* bahwa setidaknya satu sinyal *input* adalah nol (Bliesener, tanpa tahun : A-5).

Sinyal status *AND* didapat dari semua *one-bit operand*. Elemen *AND* juga dapat digunakan untuk *multibit operand*. Namun, fungsi blok ini harus digunakan dengan melihat *multibit operand*.



Gambar 3.11 Simbol *AND*.

#### 3.2.1.3.8.2 OR

Elemen *OR* ditunjukkan dengan persegi panjang, dengan fungsi pengidentifikasi “>=”. Elemen ini mempunyai 2-16 inputan. Sinyal *output* adalah 1, jika paling tidak satu sinyal input adalah 1. Sebuah sinyal *output* nol berarti sinyal input adalah nol (Bliesener, tanpa tahun : A-6).

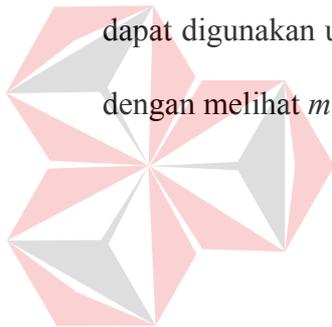
Sinyal status *OR* didapat dari semua *one-bit operand*. Elemen *OR* juga dapat digunakan untuk *multibit operand*. Namun, fungsi blok ini harus digunakan dengan melihat *multibit operand*.

Gambar 3.12 Simbol *OR*.

### 3.2.1.3.8.3 *Exclusive OR (ExOR)*

Elemen ExOR ditunjukkan dengan persegi panjang, dengan fungsi pengidentifikasi “=1”. Elemen ini mempunyai 2-16 inputan. Sinyal *output* adalah 1, jika hanya satu sinyal *input* adalah 1. Sebuah sinyal *output* nol berarti kedua sinyal *input* sama (Bliesener, tanpa tahun : A-6).

Sinyal status ExOR didapat dari 2 *one-bit operand*. Elemen ExOR juga dapat digunakan untuk *multibit operand*. Namun, fungsi blok ini harus digunakan dengan melihat *multibit operand*.

Gambar 3.13 Simbol *Exclusive OR*.

### 3.2.1.3.9 *Flip-flop*

Flip-flop mempunyai 2 elemen yang digunakan, yaitu :

#### 3.2.1.3.9.1 *S/R (Reset Dominant)*

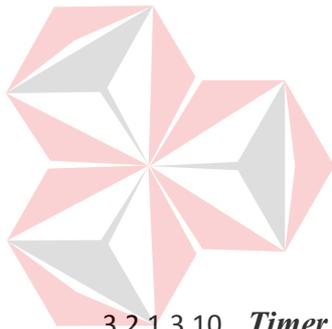
*Set input* fungsi blok ini diidentifikasi oleh S, sedangkan *reset input* diidentifikasi oleh R. disebut juga dengan *input reset* dominan. Jika kedua sinyal *input* adalah 1, maka operan *reset*. Jika hanya satu *set* sinyal input, maka operan ditetapkan *retentively* sampai *reset* satu sinyal pada *input reset* (Bliesener, tanpa tahun : A-7).



Gambar 3.14 Simbol S/R (*Reset Dominant*).

#### 3.2.1.3.9.2 R/S (*Set Dominant*)

*Set input* fungsi blok ini diidentifikasi oleh S, sedangkan *reset input* diidentifikasi oleh R. disebut juga dengan *input set* dominan. Jika kedua sinyal *input* adalah 1, maka operan ditetapkan *retentively*. Jika hanya satu sinyal *input* diulang sekali, maka operan *reset* sampai diatur lagi oleh satu sinyal dari inputan yang telah ditetapkan (Bliesener, tanpa tahun : A-7).



Gambar 3.15 Simbol R/S (*Set Dominant*).

#### 3.2.1.3.10 *Timer*

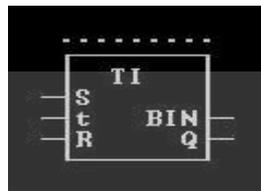
*Timer* berfungsi untuk mengaktifkan suatu keluaran dengan *interval* waktu yang dapat diatur. Pengaturan waktu dilakukan melalui nilai *setting* (*preset value*). *Timer* tersebut akan bekerja bila diberi *input* dan mendapat pulsa *clock*. Untuk pulsa *clock* sudah disediakan oleh pembuat PLC. Besarnya nilai pulsa *clock* pada setiap *timer* tergantung pada nomor timer yang digunakan. Saat *input timer ON* maka *timer* mulai mencacah pulsa dari 0 sampai *preset value*. Bila sudah mencapai *preset value* maka akan mengaktifkan *output* yang telah ditentukan (Himpunan Mahasiswa Elektro Universitas Satyagama, 2001 : 9). Untuk mengatur timer ada 3 yang harus diperhatikan, yaitu : *timer status* (T) digunakan

untuk *On/Off*, *timer preset* (TP) berisi jumlah maksimal dari *timer*. *Timer word* (TW) berisi nilai aktual dari *timer*.

*Timer* pada *function chart* mempunyai 3 *input* dan 2 *output*. Pada *function chart* memiliki 3 jenis timer, yaitu :

### 3.2.1.3.10.1 Pulse Timer

*Pulse timer* ditunjukkan dengan fungsi pengidentifikasi “T1”. *Pulse timer* berakhir ketika sinyal input pada “S” adalah 0. Saat *timer* sedang berjalan, keluaran status *timer* pada “Q” adalah satu sinyal (Bliesener, tanpa tahun : A-8).



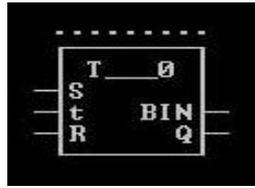
Gambar 3.16 Simbol *Pulse Timer*.

S	Digunakan untuk <i>load</i> sinyal positif TP ( <i>Timer Preset</i> ) ke dalam TW ( <i>Timer Word</i> ) dan menjalankan <i>timer</i> (Status timer = 1).
T	Nilai masukkan <i>multibit</i> diload ke TP dengan sinyal positif.
R	Reset sinyal positif status <i>timer</i> , kemudian TW dibuat menjadi 0 lagi.
BIN	Keluaran nilai TW.
Q	Keluaran status <i>timer</i> .

Tabel 3.2 Keterangan Simbol *Pulse Timer*.

### 3.2.1.3.10.2 Timer Switch-On Delay

*Timer Switch-On delay* ditunjukkan dengan fungsi pengidentifikasi “T<sub>0</sub>”. *Timer* hanya berjalan selama sinyal pada *input set* “S” adalah 1. Status *timer* pada *output* “Q” tidak 1 sampai *timer* telah berakhir (Bliesener, tanpa tahun : A-9).

Gambar 3.17 Simbol Timer *Switch-On Delay*.

S	Sinyal positif pada saat timer distart.
T	Nilai masukkan multibit di-load ke TP (Timer Preset) dan TW (Timer Word), ketika sinyal terdeteksi 0 pada inputan "S"
R	Sebuah sinyal input direset (T=0 dan TW=0).
BIN	Keluaran nilai TW.
Q	Keluaran status timer.

Tabel 3.3 Keterangan Simbol Timer *Switch-On Delay*.

### 3.2.1.3.10.3 Timer *Switch-Off Delay*

Timer *Switch-Off delay* ditunjukkan dengan fungsi pengidentifikasi "0\_T".

Timer hanya berjalan selama sinyal pada *input set* "S" adalah 0. Setelah *timer* berakhir, status *timer* pada *output* "Q" adalah 0 (Bliesener, tanpa tahun : A-9).

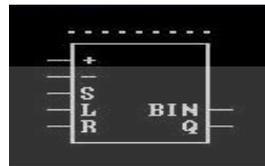
Gambar 3.18 Simbol Timer *Switch-off Delay*.

S	Sinyal positif pada saat timer distart.
t	Nilai masukkan multibit di-load ke TP (Timer Preset) dan TW (Timer Word), ketika sinyal terdeteksi 1 pada inputan "S"
R	Sebuah sinyal input timer status (T) direset.
BIN	Keluaran nilai TW.
Q	Keluaran status timer.

Tabel 3.4 Keterangan Simbol Timer *Switch-Off Delay*.

### 3.2.1.3.11 Counter

Fungsi *counter* adalah mencacah pulsa yang masuk. Sepintas cara kerja *counter* dan *timer* mirip. Perbedaannya adalah *timer* mencacah pulsa *internal* sedangkan *counter* mencacah pulsa dari luar (Himpunan Mahasiswa Elektro Universitas Satyagama, 2001 : 9). Untuk mengatur *counter* ada 3 yang harus diperhatikan, yaitu : *counter status* (C) digunakan untuk *On/Off*, *counter preset* (CP) berisi jumlah maksimal dari *counter*. *counter word* (CW) berisi nilai aktual dari *counter*. Counter pada *function chart* mempunyai 5 *input* dan 2 *output*.



Gambar 3.19 Simbol *Counter*.

+	Digunakan untuk meningkatkan nilai CW (Counter Word). Jika CW telah mencapai Cp (Counter Preset) atau CW=CP, counter akan reset
-	Digunakan untuk mengurangi nilai CW. Jika CW adalah 0 (CW=0). Counter akan reset.
S	Untuk <i>meload</i> sinyal positif dari CP ke CW dan mengaktifkan status counter (C=1).
L	Nilai multibit diload ke CP ketika sinyal positif terdeteksi pada input "S"
R	Untuk <i>mereset</i> status counter (C).
BIN	Keluaran nilai CW.

Tabel 3.5 Keterangan Simbol *Counter*.

### 3.2.1.3.12 *Special Assignments*

*Special assignments* mempunyai 2 elemen yang digunakan, yaitu :

#### 3.2.1.3.12.1 *One-bit Assignment*

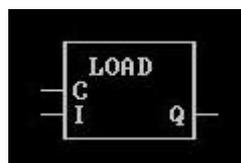
*One-bit assignment* ditunjukkan dengan fungsi pengidentifikasi “=”. Logika pada sinyal *input* bertugas untuk *output*. sebuah operan dimasukkan sebagai *network input* juga dapat koneksi ke *network output directly* dengan sambungan *wire*. "*one bit assignment*" fungsi blok memungkinkan untuk meniadakan dan menetapkan sinyal logis *one-bit* operan. Negasi ini memerlukan sinyal pada masukan atau keluaran dari *assignment* (Bliesener, tanpa tahun : A-11).



Gambar 3.20 Simbol *One-bit Assignment*.

#### 3.2.1.3.12.2 *Multibit Assignment*

*Multibit Assignment* ditunjukkan dengan fungsi pengidentifikasi “LOAD”. Nilai pada *input* satu diberikan ketika sinyal positif terdeteksi dimasukkan ke “C”. sebuah operan dimasukkan sebagai *network output*, juga dapat dihubungkan langsung ke *network output* dengan *wire connection*. Kemudian berlangsung independen dari ujung sinyal (Bliesener, tanpa tahun : A-11).



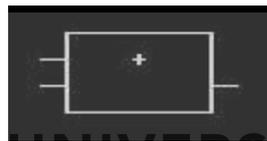
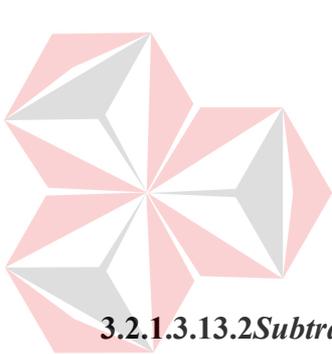
Gambar 3.21 Simbol *Multibit Assignment*.

### 3.2.1.3.13 *Arithmetic Operations*

Pada *function chart* terdapat 4 elemen yang digunakan, yaitu :

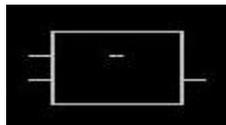
#### 3.2.1.3.13.1 *Addition*

Fungsi blok ini adalah ditulis sebagai persegi panjang dengan fungsi pengidenfikasi “+”. Hingga 16 *multibit* dapat nilai tambah per-fungsi blok. Hasil tambahan ditugaskan untuk *output*. Operan *multibit* apapun dapat ditambahkan (Bliesener, tanpa tahun : A-12).

Gambar 3.22 Simbol *Addition*.

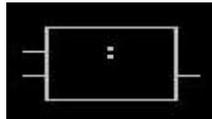
#### 3.2.1.3.13.2 *Subtraction*

Fungsi blok ini adalah ditulis sebagai persegi panjang dengan fungsi pengidenfikasi “-”. Nilai-nilai *multibit* yang di input ditafsirkan sebagai keseluruhan, bawah nilai *multibit* dikurangi dari atas nilai *multibit*. Hasil pengurangan ditugaskan untuk *output* (Bliesener, tanpa tahun : A-12).

Gambar 3.23 Simbol *Subtraction*.

### 3.2.1.3.13.3 Division

Fungsi blok ini adalah ditulis sebagai persegi panjang dengan fungsi pengidentifikasi “:”. Nilai *multibit* atas dibagi dengan nilai *multibit* rendah. Pembagian hasil ditugaskan untuk *output* (Bliesener, tanpa tahun : A-13).



Gambar 3.24 Simbol *Division*.

### 3.2.1.3.13.4 Multiplication

Fungsi blok ini adalah ditulis sebagai persegi panjang dengan fungsi pengidentifikasi “\*”. Bilangan bulat kedua nilai *multibit* dikalikan. Hasil perkalian ditugaskan untuk *output* (Bliesener, tanpa tahun : A-13).



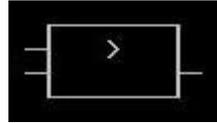
Gambar 3.25 Simbol *Multiplication*.

### 3.2.1.3.14 Comparison Operation

*Comparison operation* pada *function chart* ada 6, yaitu :

#### 3.2.1.3.14.1 Greater Than

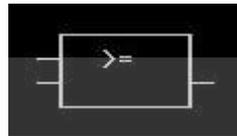
Fungsi blok ini ditunjukkan oleh fungsi pengidentifikasi “>”. Sinyal *Output* adalah 1, jika nilai *multibit* diinput lebih besar daripada nilai *multibit* dibawah inputan (Bliesener, tanpa tahun : A-14).



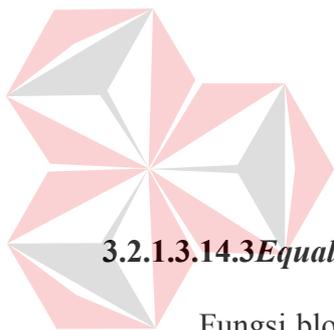
Gambar 3.26 Simbol *Greater Than*.

#### 3.2.1.3.14.2 *Equal or Greater Than*

Fungsi blok ini ditunjukkan oleh fungsi pengidentifikasi “>=”. Sinyal *output* adalah 1, jika nilai *multibit* diinput atas lebih besar dari atau sama dengan nilai *multibit* diinput lebih rendah (Bliesener, tanpa tahun : A-14).

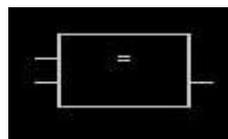


Gambar 3.27 Simbol *Equal or Greater Than*.



#### 3.2.1.3.14.3 *Equal*

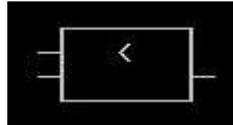
Fungsi blok ini ditunjukkan oleh fungsi pengidentifikasi “=” . Sinyal *output* adalah 1 jika kedua nilai *multibit* sama (Bliesener, tanpa tahun : A-15).



Gambar 3.28 Simbol *Equal*.

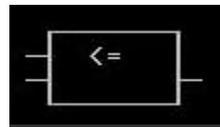
#### 3.2.1.3.14.4 *Less Than*

Fungsi blok ini ditunjukkan oleh fungsi pengidentifikasi “<”. Sinyal *output* adalah 1 jika nilai *multibit* diinput atas kurang dari nilai *multibit* diinput yang lebih rendah (Bliesener, tanpa tahun : A-15).

Gambar 3.29 Simbol *Less Than*.

### 3.2.1.3.14.5 *Equal or Less Than*

Fungsi blok ini ditunjukkan oleh fungsi pengidentifikasi “ $\leq$ ”. Sinyal *output* adalah 1 jika nilai pada *multibit input* atas kurang dari atau sama dengan nilai *multibit* diinput rendah (Bliesener, tanpa tahun : A-15).

Gambar 3.30 Simbol *Equal or Less Than*.

### 3.2.1.3.14.6 *Not Equal*

Fungsi blok ini ditunjukkan oleh fungsi pengidentifikasi “ $\neq$ ”. Sinyal *output* adalah 1 jika nilai *multibit* berbeda (Bliesener, tanpa tahun : A-15).

Gambar 3.31 Simbol *Not Equal*.

Untuk *comparison operation* nilai *multibit* ditafsirkan sebagai bilangan bulat positif (*range* nilai antara 0 hingga 65535).

Sumber : Bliesener, M., dkk. tanpa tahun. *Festo Software-tools Function Chart FPC 100*.

### 3.2.1.4 Draw Wire

*Command draw wire* berfungsi untuk menghubungkan antara elemen satu dengan elemen lainnya. Posisi kursor digunakan untuk menentukan awal *wire*. Poin *wire* dan ujung dari *wire*. Awal dan akhir dari *wire* dapat dilihat pada Tabel 3.6.

<b>Start</b>	<b>End</b>
<i>Network Input</i>	<i>Network Output</i>
<i>Intermediate</i>	<i>Intermediate</i>
Output dari Fungsi Blok	Input dari Fungsi Blok
<i>Wire Knot</i>	<i>Wire Knot</i>
<i>String Parameter</i>	<i>Jump</i>

Tabel 3.6 *Start* dan *End* dari *Wire*.

### 3.2.1.5 Modify/Enter

*Command* ini berfungsi untuk memberikan atau mengedit nama pada elemen, dengan cara menekan F3, kemudian tuliskan nama, lalu tekan enter.

### 3.2.1.6 Shift Element

*Command* ini berfungsi untuk mengatur letak elemen. Menggunakan *command* ini dengan cara : meletakkan kursor pada elemen yang ingin dipindah, tekan F4. Setelah tekan F4 akan muncul seperti pada Gambar 3.32.



Gambar 3.32 Tampilan *Command Shift Element*.

Arah pindahannya dapat keatas (F1), kebawah (F2), kekiri (F3), atau kekanan (F4). Untuk memindahkannya tidak hanya dapat menggunakan tombol-tombol tersebut, tetapi dapat juga menggunakan tombol panah pada *keyboard*. Setelah menentukan letaknya, selanjutnya tekan F8 atau Esc.

### 3.2.1.7 Delete Element

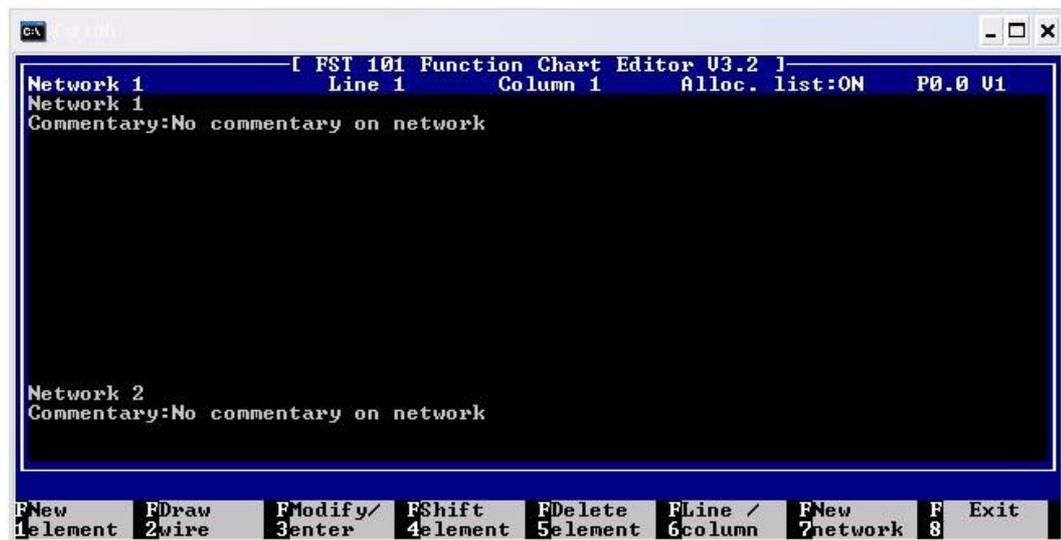
*Command* ini digunakan untuk menghapus elemen, dengan cara meletakkan kursor ke elemen yang akan dihapus, kemudian tekan F5.

### 3.2.1.8 Line/Coloum

*Command* ini digunakan untuk menambahkan atau menghapus baris/kolom, dengan cara menekan F6.

### 3.2.1.9 New Network

*Command* ini digunakan untuk menambahkan *network* baru, dengan cara menekan F7.



```

FST 101 Function Chart Editor U3.2
Network 1      Line 1      Column 1      Alloc. list:0N      P0.0 U1
Network 1
Commentary:No commentary on network

Network 2
Commentary:No commentary on network

F1 New Element  F2 Draw Wire  F3 Modify/Enter  F4 Shift Element  F5 Delete Element  F6 Line / Column  F7 New Network  F8 Exit
  
```

Gambar 3.33 Tampilan *Command New Network*.

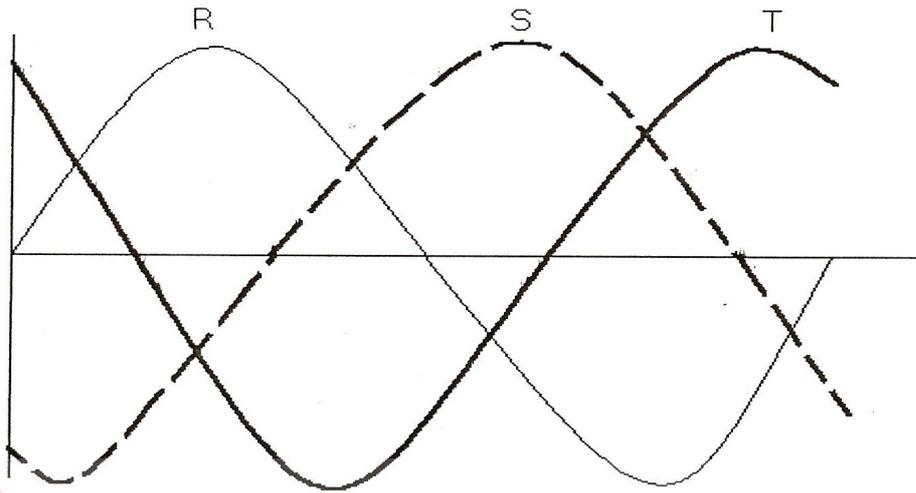
#### 3.2.1.10 Exit

*Command* ini untuk keluar dari menu *edit command* atau keluar dari program dengan menekan F8.

### 3.3 Motor Tiga Fasa

Motor tiga fasa adalah suatu motor AC yang menggunakan suplay tegangan tiga fasa. Dimana tegangan AC tiga fasa memiliki 4 hantaran dimana dari keempat hantaran tersebut memiliki tiga fasa yang diberi nama R, S, T, dan satu hantaran netral.

Pada tegangan tiga fasa memiliki beda fasa dari R, S, maupun pada T yang dapat mengakibatkan perputaran pada motor (Ardiono, 2004 : 17). Untuk memperjelas dapat dilihat pada Gambar 3.34.



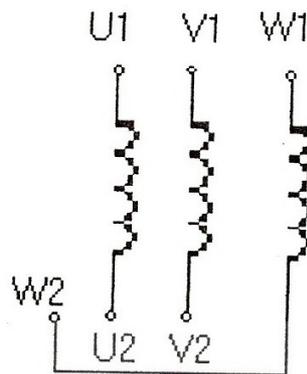
Gambar 3.34 Beda Fasa pada RST.

Tujuan penggunaan motor tiga fasa ini yaitu agar sistem lebih stabil, karena motor tiga fasa lebih stabil jika di bandingkan dengan motor AC satu fasa, motor tiga fasa juga memiliki torsi yang lebih besar.

Sumber : Ardiono, Rachman, dkk. 2004. *Laporan Kerja Praktek Di PT. FESTO Surabaya.*

### 3.3.1 Struktur Motor Tiga Fasa

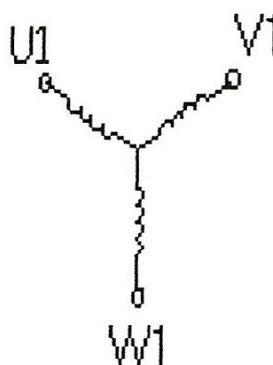
Didalam motor tiga fasa terdapat tiga lilitan dimana dari ketiga lilitan tadi terdapat 6 buah hantaran yang terjadi jadi dua *group* yaitu U1, V1, W1 dan U2, V2, W2 untuk lebih jelasnya lihat Gambar 3.35.



Gambar 3.35 Struktur Motor Tiga Fasa.

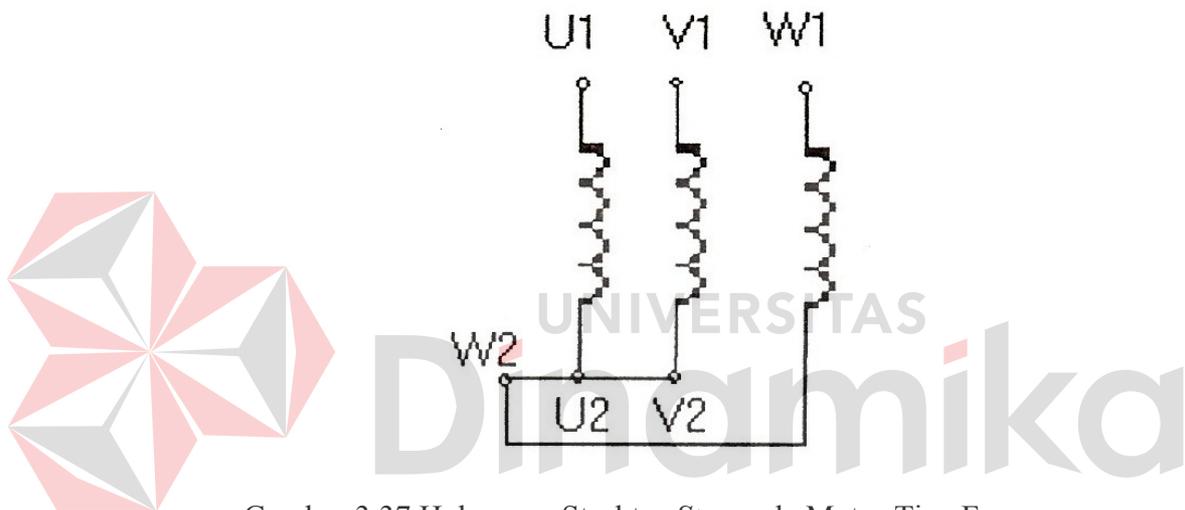
Pada motor tiga fasa terdapat beberapa cara untuk mengendalikan motor tiga fasa diantaranya dengan menggunakan struktur *star* dan delta. Keduanya memiliki kelebihan dan kekurangan, diantaranya :

- a. Pada struktur *star* memiliki kelebihan arusnya lebih kecil jika dibandingkan dengan delta namun pada struktur *star* motor akan lebih lama untuk mencapai kestabilan. Oleh karena itu, *star* banyak digunakan pada sistem yang membutuhkan kestabilan dalam waktu singkat. Jika suplay tegangan yang digunakan adalah 220/380 maka struktur *star* akan dipakai untuk tegangan 380 volt. Hubungan dari struktur *star* adalah seperti Gambar 3.36.



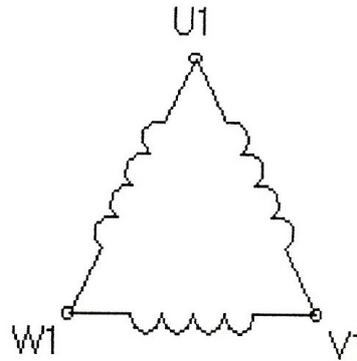
Gambar 3.36 Struktur *Star*.

Jadi hubungan untuk struktur *star* pada motor tiga fasa tampak seperti pada Gambar 3.37.

Gambar 3.37 Hubungan Struktur *Star* pada Motor Tiga Fasa.

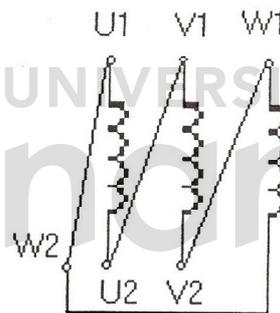
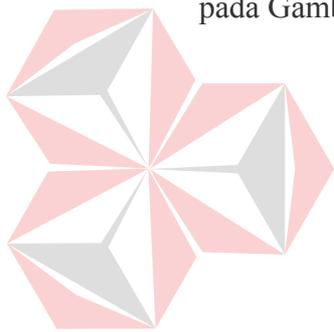
- b. Sedangkan pada struktur delta biasanya digunakan untuk motor-motor dengan arus besar. Dengan menggunakan struktur delta, start pada motor lebih cepat sehingga kestabilan pada putaran motor akan lebih cepat dicapai dibandingkan dengan struktur *star*, namun pada struktur delta dibutuhkan arus yang lebih besar daripada struktur *star*. Oleh karena itu, pada sebagian motor digunakan keduanya yaitu *Star* untuk mendapatkan kestabilan motor yang cepat, kemudian delta agar arus yang terpakai kecil. Jika suplay tegangan yang digunakan adalah 220/380 maka struktur delta

akan dipakai untuk tegangan 220 volt. Hubungan dari struktur delta adalah seperti pada Gambar 3.38.



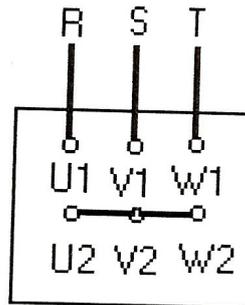
Gambar 3.38 Struktur Delta.

Jadi hubungan untuk struktur delta pada motor tiga fasa tampak seperti pada Gambar 3.39.



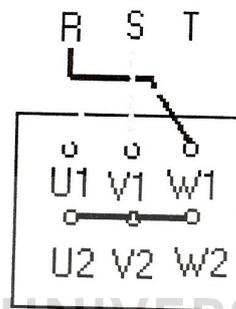
Gambar 3.39 Hubungan Struktur Delta pada Motor Tiga Fasa.

Pada motor AC tiga fasa arah putaran motor (rotasi) dapat dirubah dengan mengubah posisi R dan T. Jika R dihubungkan dengan U1 dan S dengan V1 dan T dengan W1 maka motor akan berputar ke kanan. Hubungan listriknya tampak pada Gambar 3.40.

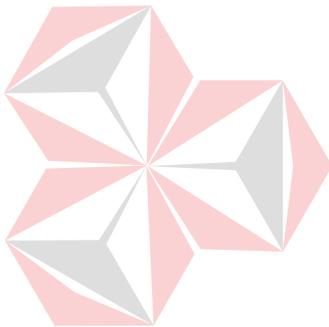


Gambar 3.40 Rangkaian Pembalik Arah Kekanan.

Namun jika posisi R dan T dibalik dimana R dihubungkan dengan W1 dan T dengan U1 maka motor akan berputar kekiri.



Gambar 3.41 Rangkaian Pembalik Arah Kekiri.



Sumber : Ardiono, Rachman, dkk. 2004. *Laporan Kerja Praktek Di PT. FESTO Surabaya.*

### 3.3.2 Fungsi Motor Tiga Fasa

Motor tiga fasa digunakan sebagai penggerak, yaitu memindahkan benda atau barang dari satu tempat ke tempat lain melalui *conveyor (belt conveyor)*.

### 3.4 *Inverter*

*Inverter* adalah peralatan elektronik yang melakukan konfersi arus dari arus searah ke arus bolak-balik. Tegangan dan frekuensi arus hasil konfersi tergantung dari transformator dan alat kontrol yang digunakan. Penggunaan *inverter* antara lain sebagai berikut :

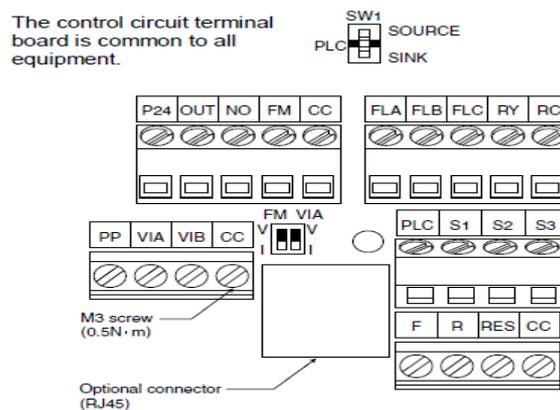
- *DC power utilitation.*
- *Uninterruptible Power Supplies (UPS).*
- *Air Conditioning.*
- Menjalankan motor AC, dll.

Pada kerja praktek ini aplikasi *inverter* yang diterapkan adalah untuk menjalankan motor AC. Pengontrolan motor AC tidak sama dengan pengontrolan motor DC. Pada motor DC untuk merubah arah putaran motor kita cukup merubah polaritas tegangan pada motor. Sedangkan untuk mengatur kecepatan motor DC dapat dilakukan dengan cara memberi sinyal *Pulse Width Modulation* (PWM), yaitu suatu sinyal yang mempunyai lebar data *high* dan *low* yang berbeda. Metode pengontrolan pada motor DC tidak dapat digunakan pada motor AC. Pengaturan arah putaran motor AC tiga fasa dapat dilakukan seperti yang telah dibahas pada subbab 3.3.1, sedangkan untuk pengaturan kecepatannya dilakukan dengan dengan cara yang lebih rumit, dibandingkan dengan memberikan sinyal PWM untuk motor DC. Dengan menggunakan *inverter*, kita dapat merubah arah putaran, merubah kecepatan, melakukan *emergency break*, dsb. pada motor AC hanya dengan memberikan sinyal inputan yang sesuai pada *inverter*.

### 3.4.1 Koneksi

Pada *terminal board* terdapat *terminal-terminal* yang digunakan untuk memberikan sinyal masukan yang diperlukan untuk pengontrolan *inverter*.

Gambar dari *terminal board* seperti Gambar 3.42 dibawah ini :



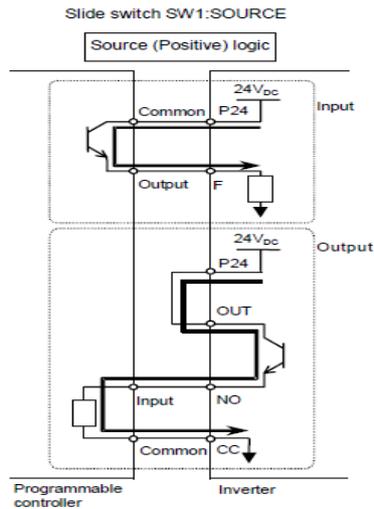
Gambar 3.42 Terminal Board.

Untuk kegunaan dari *terminal-terminal* yang ada dapat dilihat dalam lampiran 4. Kegunaan SW1 adalah untuk memilih mode yang digunakan pada saat mengoperasikan *inverter*, yaitu mode *source*, PLC, dan *sink*.

### 3.4.2 Mode Operasi Inverter

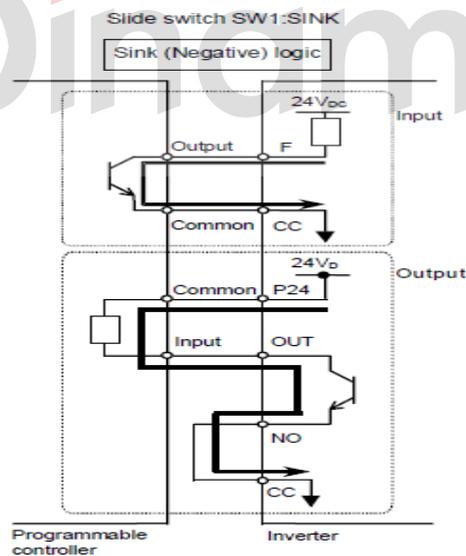
#### 3.4.2.1 Mode Source

Pada mode operasi ini sinyal inputan diberikan pada *terminal* masukan dengan cara menghubungkannya dengan terminal P24. Koneksi mode ini dapat dilihat pada Gambar 3.43 dibawah ini:

Gambar 3.43 Mode *Source*.

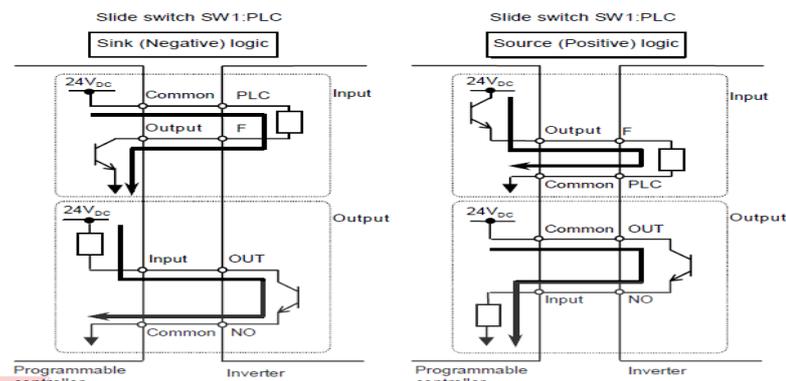
### 3.4.2.2 Mode *Sink*

Pada mode operasi ini sinyal inputan diberikan pada *terminal* masukan dengan cara menghubungkannya dengan terminal CC. Koneksi mode ini dapat dilihat pada Gambar 3.44 dibawah ini :

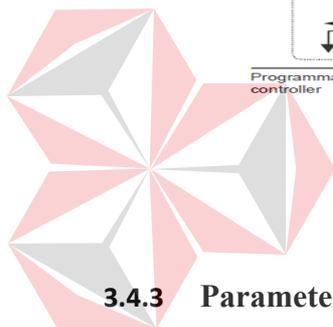
Gambar 3.44 Mode *Sink*.

### 3.4.2.3 Mode PLC

Pada mode operasi ini sinyal inputan diberikan pada *terminal* masukan dengan cara menghubungkan *ground* PLC dengan terminal PLC, lalu menghubungkan terminal masukan dengan *output* dari PLC. Koneksi mode ini dapat dilihat pada Gambar 3.45 dibawah ini :



Gambar 3.45 Mode PLC.



### 3.4.3 Parameter Inverter

Parameter yang digunakan pada kerja praktek ini adalah sebagai berikut :

- CNOD.
- FNOD.
- SR1-SR7.
- F287-F294.

Untuk mode-mode yang lain dapat dilihat di Lampiran 5.

### 3.4.3.1 CNOD

CNOD adalah parameter yang digunakan sebagai *command mode selection*. Sinyal-sinyal masukan ke *inverter* dapat dikirimkan melalui dua cara yaitu, melalui *terminal board* atau melalui *operation panel*.

Untuk merubah arah putaran motor AC tidak dapat dilakukan melalui *operation panel*. Sedangkan untuk operasi melalui *terminal board* kita dapat merubah kecepatan motor, merubah arah putaran motor, mengatur control PID, dll.

Title	Function	Adjustment range	Default setting
<i>CNOD</i>	Command mode selection	0: Terminal board 1: Panel	1
<i>FNOD</i>	Frequency setting mode	0: Internal potentiometer setting 1: VIA 2: VIB 3: Operation panel 4: Serial communication 5: External contact up/down 6: VIA+VIB (Override)	0

Tabel 3.7 CNOD.

### 3.4.3.2 FNOD

FNOD digunakan untuk menentukan *frequency setting mode*. Pengaturan frekuensi dapat dilakukan dengan tujuh cara yaitu :

- Potensiometer pada *operation panel*.
- VIA.
- VIB.
- Tombol atas dan bawah pada *operation panel*.
- Komunikasi serial.
- Saklar *eksternal*.
- VIA dan VIB.

#### 3.4.3.2.1 **Potensiometer Pada *Operation Panel***

Cara ini dapat dipilih dengan cara memberikan nilai 0 pada parameter FNOD. Pengaturan frekuensi dilakukan dengan cara memutar potensiometer yang terdapat pada *operation panel*.

#### 3.4.3.2.2 **VIA**

Cara ini dapat dipilih dengan cara memberikan nilai 1 pada parameter FNOD. Pengaturan frekuensi dilakukan dengan cara memberikan tegangan 0-10Vdc atau 4-20mAdc pada *terminal* VIA.

#### 3.4.3.2.3 **VIB**

Cara ini dapat dipilih dengan cara memberikan nilai 2 pada parameter FNOD. Pengaturan frekuensi dilakukan dengan cara memberikan tegangan 0-10Vdc pada *terminal* VIB.

#### 3.4.3.2.4 **Tombol Atas dan Bawah Pada *Operation Panel***

Cara ini dapat dipilih dengan cara memberikan nilai 3 pada parameter FNOD. Pengaturan frekuensi dapat dilakukan dengan cara menekan tombol atas dan bawah yang terdapat pada *operation panel*.

#### 3.4.3.2.5 **Komunikasi Serial**

Cara ini dapat dipilih dengan cara memberikan nilai 4 pada parameter FNOD. Pengaturan frekuensi dapat dilakukan dengan cara komunikasi serial.

Komunikasi serial dapat dilakukan dengan komputer ataupun dengan mikrokontroler.

#### 3.4.3.2.6 Saklar *Eksternal*

Cara ini dapat dipilih dengan cara memberikan nilai 5 pada parameter FNOD. Pengaturan dapat dilakukan dengan cara memberi inputan pada *terminal* S1, S2, S3, dan RES. Terminal-terminal tersebut digunakan sebagai selektor kecepatan (total ada 16 *presets* kecepatan). S1 berfungsi sebagai LSB dan RES berfungsi sebagai MSB.

#### 3.4.3.2.7 VIA dan VIB

Cara ini dapat dipilih dengan cara memberikan nilai 6 pada parameter FNOD. Pengaturan frekuensi dilakukan dengan cara menjumlah nilai yang masuk pada *terminal* VIA dan VIB.

#### 3.4.3.3 SR1-SR7

Parameter ini digunakan untuk memberi nilai pada *presets* kecepatan satu sampai tujuh.

#### 3.4.3.4 F287-F294

Parameter ini digunakan untuk member nilai pada *presets* kecepatan delapan sampai lima belas. Untuk dapat memakai *presets* delapan sampai lima belas, kita harus memberi nilai 9 pada parameter F113.

## BAB IV

### METODE KERJA PRAKTEK

#### 4.1 Metode Kerja Praktek

Metode yang digunakan dalam pengerjaan kerja praktek ini adalah seperti

Gambar 4.1 dibawah ini :



Gambar 4.1 Alur Metode Kerja Praktek.

### 4.1.1 Pengumpulan Informasi

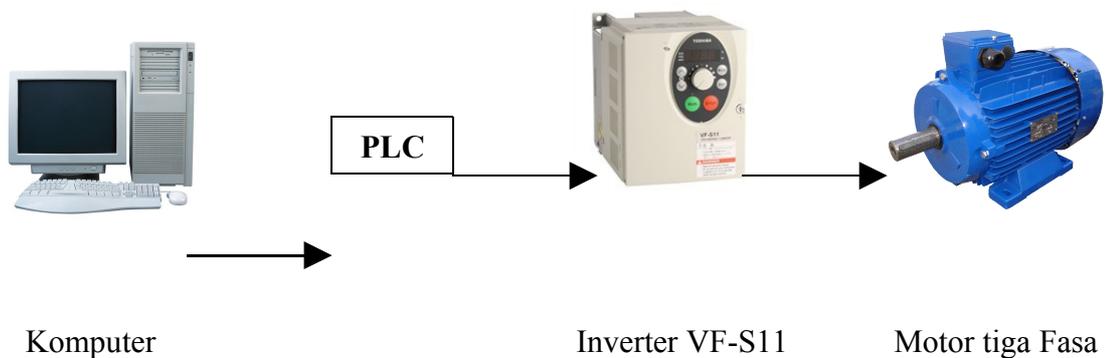
Kerja praktek pertama-tama kami lakukan dengan pengumpulan informasi melalui wawancara. Disini kami mencoba menggali kebutuhan dengan bertanya langsung kepada asisten laboratorium, pembimbing kerja praktek, dan penyelia untuk mendapatkan data-data serta informasi yang berhubungan dengan objek kerja praktek.

### 4.1.2 Studi Literatur

Kami mempelajari dan membaca buku, maupun literatur lainnya yang berkaitan dengan objek kerja praktek termasuk permasalahan yang dihadapi.

### 4.1.3 Perancangan *Hardware*

Perancangan *hardware* ini merupakan blok diagram implementasi dari tugas kerja praktek kami. Blok diagram tersebut ditunjukkan oleh Gambar 4.2 dibawah ini:



Gambar 4.2 Blok Diagram Perancangan *Hardware*.

#### 4.1.4 Perancangan Perangkat Lunak (*Software*)

Perancangan perangkat lunak dikerjakan pada *software* untuk pemrograman PLC yaitu FST100 *software*, dimana di dalamnya terdapat beberapa bahasa pemrograman yang dapat dijalankan oleh PLC dengan berbagai tipe yang telah disediakan oleh *software* tersebut.

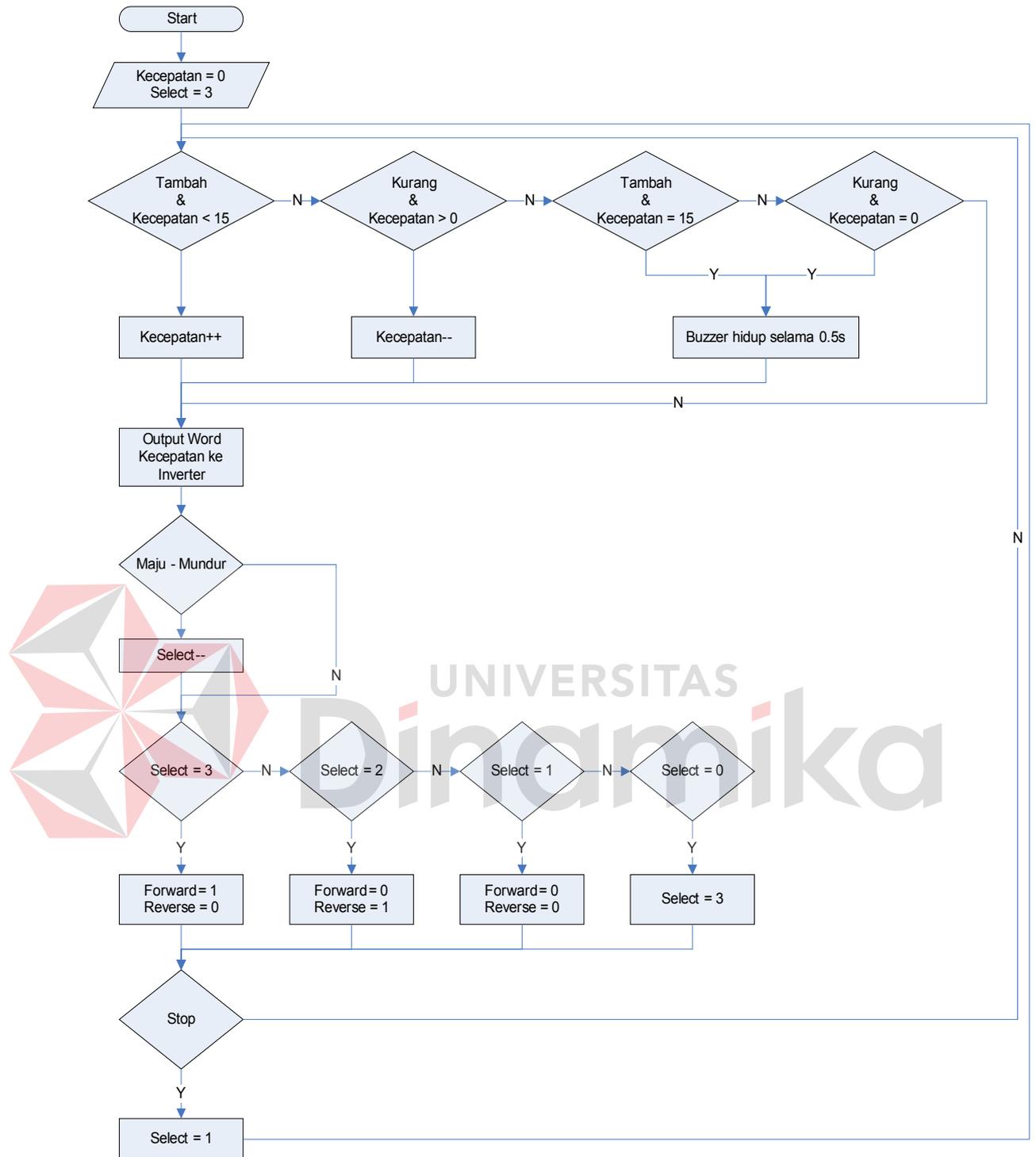
Untuk perancangan dan pembuatan perangkat lunak diperlukan *flowchart* (diagram alir). Pembuatan *flowchart* digunakan untuk mempermudah dalam pembuatan program dan dapat dimengerti dengan mudah oleh orang awam. Pembuatan sistem menjelaskan tentang *flowchart* untuk program pada PLC secara umum dan dilanjutkan dengan menjelaskan tiap proses yang ada dalam *flowchart* tersebut.



UNIVERSITAS  
**Dinamika**

##### 4.1.4.1 Perancangan Perangkat Lunak pada PLC

Gambar 4.3 menjelaskan tentang alur program yang akan di *download* kedalam PLC. Didalam program terdapat 4 tombol yang digunakan sebagai *input*. Dimana tombol1 (Tambah) digunakan untuk menambahkan kecepatan motor yang berada pada alamat I0.2, tombol2 (Kurang) digunakan untuk mengurangi kecepatan motor yang berada pada alamat I0.3, tombol3 (MaMu) digunakan untuk mengubah arah putar motor (maju-mundur) yang berada pada alamat I0.0, dan tombol4 (Stop) digunakan untuk menghentikan putaran motor sewaktu-waktu yang berada pada alamat I0.1.

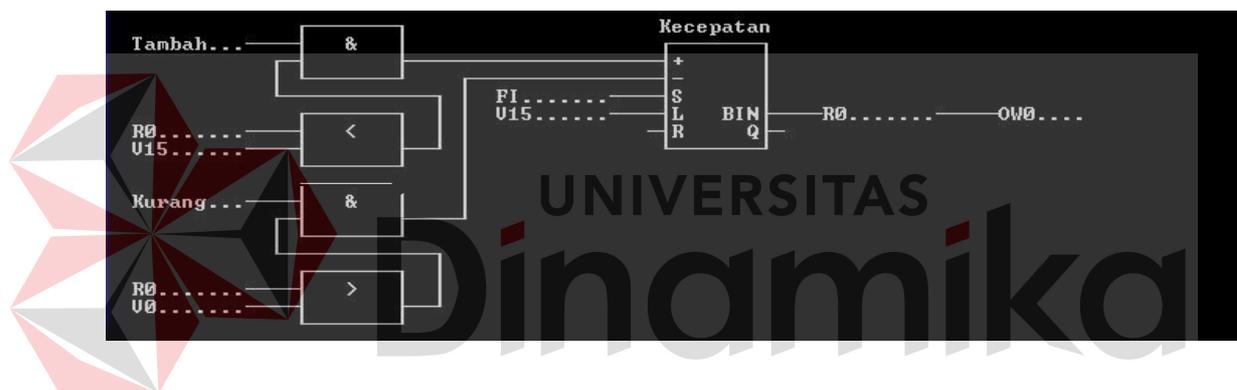


Gambar 4.3 *Flowchart* Perangkat Lunak pada PLC.

Berdasarkan Gambar 4.3 dapat dibuat program seperti pada Lampiran 3.

Program pada Lampiran 3. digunakan pada PLC untuk menerima masukan dari alamat I0.0-I0.3, untuk mengaktifkan *Inverter* dari keluaran PLC pada alamat O0.0-O0.3 dan O1.0-O1.1. Outputan dari *Inverter* akan megerakkan motor tiga fasa sesuai dengan data pada program FST. Jika kecepatan motor sudah sampai maksimal dan minimal maka buzzer akan aktif. Alamat buzzer berada di O1.5.

#### 4.1.4.2 Cara Kerja Program



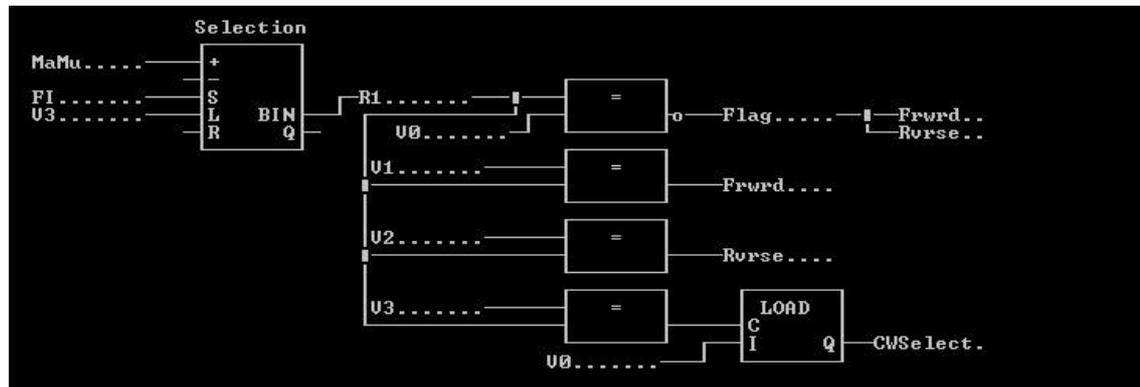
Gambar 4.4 *Network 1*.

Potongan program pada Gambar 4.4 berjalan dengan alur sebagai berikut :

1. Pada awal program berjalan, *set counter* Kecepatan dengan nilai *preset* 15.
2. Lakukan deteksi penekanan tombol Tambah dan tombol Kurang.
3. Jika tombol Tambah ditekan maka cek apakah nilai R0 kurang dari 15. Jika ya, tambah nilai *counter* Kecepatan dengan 1. Lanjut ke langkah 5.
4. Jika tombol Kurang ditekan maka cek apakah nilai R0 lebih dari 0. Jika ya, kurangi nilai *counter* Kecepatan dengan 1. Lanjut ke langkah 5.

5. Outputkan nilai *counter* Kecepatan pada *Output Word 0* PLC (OW0).

Dengan langkah 3 dan 4 maka nilai *counter* tidak akan keluar dari kisaran angka 0 sampai 15. Dengan kata lain, *status counter* tidak akan pernah *reset*.



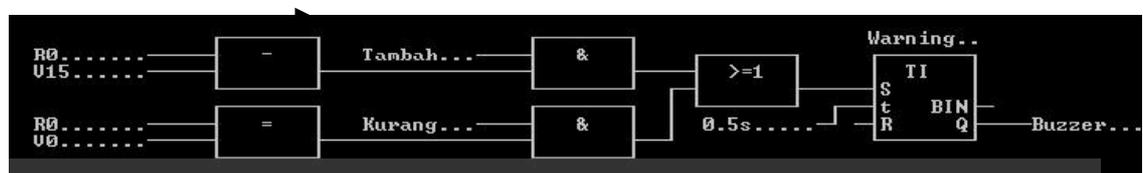
Gambar 4.5 *Network 2*.

Potongan program pada Gambar 4.5 berjalan dengan alur sebagai berikut :

1. Pada awal program berjalan , *set counter Selection* dengan nilai *preset 3*.
2. Jika ada penekanan tombol MaMu maka tambah nilai *counter Selection* dengan 1.
3. Deteksi apakah nilai *Selection* sama dengan 0. Jika ya maka nyalakan *output Frwd* dan *Rvrse* pada PLC, kembali ke langkah 2. Jika tidak, lanjut ke langkah 4.
4. Deteksi apakah nilai *Selection* sama dengan 1. Jika ya maka nyalakan *output Frwd* pada PLC, kembali ke langkah 2. Jika tidak, lanjut ke langkah 5.
5. Deteksi apakah nilai *Selection* sama dengan 2. Jika ya maka nyalakan *output Rvrse* pada PLC, kembali ke langkah 2. Jika tidak, lanjut ke langkah 6.

6. Deteksi apakah nilai *Selection* sama dengan 3. Jika ya maka berikan nilai 0 pada *counter Selection*, kembali ke langkah 2.

Dengan langkah 6 maka nilai *Selection* tidak akan melebihi 3. Karena begitu nilai *Selection* sama dengan 3 maka nilai tersebut akan diganti dengan nilai 0. Jadi dengan program diatas, urutan pergerakan motor menurut banyaknya penekanan tombol MaMu akan berulang sebagai berikut : Berhenti – Maju – Mundur – Berhenti.



Gambar 4.6 *Network 3*.

Potongan program pada Gambar 4.6 berjalan dengan alur sebagai berikut :

1. Deteksi apakah nilai R0 sama dengan 15 atau sama dengan 0.
2. Jika nilai R0 sama dengan 15 dan dideteksi penekanan tombol Tambah, maka nyalakan *timer Warning* dengan waktu 0.5s. *Timer Warning* akan menyalakan Buzzer selama *status timer* masih *set*.
3. Jika nilai R0 sama dengan 0 dan dideteksi penekanan tombol Kurang, maka nyalakan *timer Warning* dengan waktu 0.5s. *Timer Warning* akan menyalakan Buzzer selama *status timer* masih *set*.
4. Kembali ke langkah 1.

Program pada Gambar 4.6 berfungsi untuk memberi tahu kepada *user* bahwa kecepatan motor sudah mencapai maksimal atau minimal.



Gambar 4.7 *Network 4*.

Potongan program pada Gambar 4.7 digunakan untuk mendeteksi penekanan tombol Stop. Jika tombol Stop ditekan maka berikan nilai 0 pada *counter Selection*. Dengan memberikan nilai 0 pada *counter Selection* maka akan memaksa motor untuk berhenti.

Pada pemrograman *function chart* program berjalan secara sekuensial dari *network* pertama sampai terakhir. Hanya saja program berjalan sangat cepat sehingga tidak tampak sifat sekuensialnya. Jadi dapat dianggap *network-network* diatas berjalan bersama-sama (paralel).

#### 4.1.5 Konsultasi Permasalahan

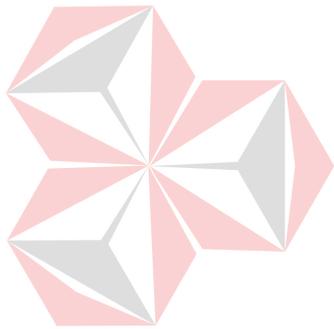
Dengan data yang terkumpul, kami berkonsultasi untuk menyelesaikan masalah kesulitan pekerjaan yang kami hadapin dan melayangkan persetujuan permohonan dana dari pihak instansi.

#### 4.1.6 Perancangan *Design*

Kami merancang dan membuat *design* untuk modul inverter. Perancangan ini bertujuan untuk mempermudah dalam pengaksesan atau pemakaian *inverter* dan menyederhanaan dalam pengkabelan.

#### 4.1.7 Pengujian Alat

Dengan selesainya perancangan *hardware* dan *software*, dilakukan uji coba. Kami memantau alat (motor tiga fasa dan *inverter*) dan mengadakan evaluasi untuk permasalahan yang timbul.



UNIVERSITAS  
**Dinamika**

## BAB V

### HASIL DAN PEMBAHASAN

Dalam percobaan ini kami menggunakan dua cara untuk menggerakkan motor tiga fasa, yaitu dengan mode *sink* (menggunakan *switch*) dan dengan mode PLC pada *inverter*.

#### 5.1 Mode *Sink*

Pada mode *sink*, hal yang pertama dilakukan adalah merubah posisi *switch* SW1 yang ada pada terminal *board inverter* ke posisi yang berlabel *sink*. Hal kedua yang dilakukan adalah menghubungkan *ground* pada kabel dengan terminal yang bertuliskan CC pada terminal *board*. Setelah kedua hal tersebut dilakukan kita sudah bisa memberikan sinyal input pada terminal-terminal *input* yang ada pada terminal *board inverter*.

Selanjutnya yang dilakukan adalah mengeset besar kecepatan pada *inverter*, seperti yang telah dijelaskan pada Bab III sebelumnya. Untuk menggerakkan motor menggunakan 5 *switch*, yaitu *switch1* (S4), *switch2* (S3), *switch3* (S2), *switch4* (S1) dan *switch5* (F/R). Jika keempat *switch* tersebut (S4, S3, S2, dan S1) diaktifkan semua maka kecepatan motor tiga fasa sebesar 60 Hz dan besar kecepatan motor yang lain dapat dilihat pada tabel dibawah ini :

Res (S4)	S3	S2	S1	Kecepatan (Hz)
0	0	0	0	0
0	0	0	1	4
0	0	1	0	8
0	0	1	1	12
0	1	0	0	16
0	1	0	1	20
0	1	1	0	24
0	1	1	1	28
1	0	0	0	32
1	0	0	1	36
1	0	1	0	40
1	0	1	1	44
1	1	0	0	48
1	1	0	1	52
1	1	1	0	56
1	1	1	1	60

Tabel 5.1 Besar Kecepatan Motor Tiga Fasa Pada *Inverter* VF-S11.

**Keterangan :**

0 = *Switch* tidak aktif.

1 = *Switch* aktif.

Jika *switch5* yang diaktifkan adalah F maka motor akan bergerak mundur (kekiri), sedangkan pada *switch5* yang diaktifkan adalah R maka motor akan bergerak maju (kekanan). Dengan asumsi posisi motor menghadap ke kiri seperti pada gambar dilampiran 5.

## 5.2 Mode PLC

Pada mode PLC, hal yang pertama dilakukan adalah merubah posisi *switch* SW1 yang ada pada terminal *board inverter* ke posisi yang berlabel PLC. Hal kedua yang dilakukan adalah menghubungkan *ground* yang ada pada PLC dengan *terminal* yang bertuliskan PLC pada terminal *board*. Setelah kedua hal tersebut dilakukan kita sudah tabel memberikan sinyal input pada terminal-terminal *input* yang ada pada *terminal board inverter*.

Untuk program yang ada pada PLC dapat dilihat pada Lampiran 3. *Allocation list* PLC dapat dilihat pada Tabel 5.2 dibawah ini.

Alamat I/O PLC	Keterangan		
O0.0	S1	-	Inverter
O0.1	S2	-	Inverter
O0.2	S3	-	Inverter
O0.4	Res(S4)	-	Inverter
O1.0	F	-	Inverter
O1.1	R	-	Inverter
Ground	PLC	Mode PLC	Inverter
O1.5	-	terkoneksi dengan buzzer	I/O device PLC
I0.0	Tombol1	digunakan untuk merubah arah putaran motor	I/O device PLC
I0.1	Tombol2	digunakan untuk menghentikan putaran motor	I/O device PLC
I0.2	Tombol3	digunakan untuk menambah kecepatan motor	I/O device PLC
I0.3	Tombol4	digunakan untuk mengurangi kecepatan motor	I/O device PLC

Tabel 5.2 *Allocation list* PLC.

Setelah program diload ke PLC, untuk menjalankan motor tiga fasa adalah dengan cara menekan tombol-tombol yang sudah disetting sesuai Tabel 5.2 diatas. Buzzer akan berbunyi atau aktif jika kecepatan motor sudah maksimal dan minimal.

## BAB VI

### PENUTUP

Berdasarkan pengujian pada perangkat keras dan perangkat lunak yang dipergunakan dalam tugas kerja praktek ini, maka dapat diambil kesimpulan dan saran-saran dari hasil yang diperoleh.

#### 6.1 Kesimpulan

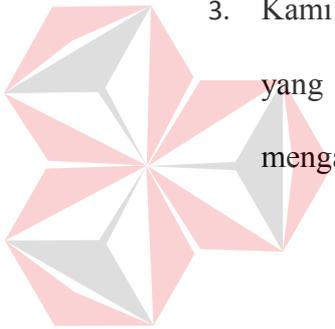
Pengaksesan motor tiga fasa melalui *inverter* yang dikontrol dengan PLC dibuat dengan hasil sebagai berikut:

1. Motor tiga fasa dapat digerakkan dengan *inverter* pada mode *sink*.
2. Motor tiga fasa dapat dirgerakkan dengan *inverter* pada mode PLC.
3. Pemrograman dengan *function chart* hampir sama dengan pemrograman *ladder diagram*. Hanya saja pada pemrograman *function chart* visualisasinya lebih mudah dipahami.
4. Cara kerja program yang dibuat dengan *function chart* adalah sekuensial, namun karena berjalan sangat cepat maka terlihat seperti berjalan secara paralel (disebut juga *quasi parallel*).
5. Dengan kami kerja praktek di Laboratorium PLC STIKOM Surabaya dapat membantu mahasiswa untuk memahami PLC dengan mudah.

## 6.2 Saran

Masih terdapat banyak kekurangan dalam penerapan kerja praktek ini, beberapa hal yang dapat dikembangkan :

1. Dalam hasil percobaan, kami masih banyak kekurangan. Diharapkan untuk percobaan menggunakan *inverter* berikutnya, dapat mengakses motor tiga fasa melalui komunikasi serial (RS232).
2. Kami berharap kepada STIKOM agar lebih meningkatkan kinerja pembelajaran tentang PLC untuk menciptakan mahasiswa yang ahli dan trampil dalam menangani PLC di dunia kerja nantinya.
3. Kami berharap kepada STIKOM agar meningkatkan mutu komputer yang ada di Laboratorium PLC untuk mendukung proses belajar mengajar dan praktikum PLC.



UNIVERSITAS  
Dinamika

**DAFTAR PUSTAKA**

Aizzue. 2009. *Belajar PLC* (Online). <http://www.scribd.com/doc/2286-9547/Belajar-PLC>, diakses tanggal 19 Januari 2010.

Ardiono, Rachman, dkk. 2004. *Laporan Kerja Praktek Di PT. FESTO Surabaya*.

Bliesener, M., dkk. tanpa tahun. *Festo Software-tools Function Chart FPC 100*.

Dodi Setyobudi, Aloysius. 1998. *Laporan Kerja Praktek di Politeknik Manufaktur* (Online). [http://www.reocities.com/al\\_dodi/kerja/KP5.pdf](http://www.reocities.com/al_dodi/kerja/KP5.pdf), diakses tanggal 19 Januari 2010.

Hidayat, Andhika. 2010. *Dasar-dasar PLC* (Online). <http://www.scribd.com/doc/25696-4898/Dasar-Dasar-PLC>, diakses tanggal 19 Januari 2010.

Himpunan Mahasiswa Elektro Universitas Satyagama. 2001. *Pengenalan Programmable Logic Controller (PLC)* (Online). <http://www.migas-indonesia.com/files/article/pengenalan-plc.doc>, diakses 2010.

Leogenda, Benyamin dan Benny P. 2009. *Analisa Programmable Logic Control PT. Gucci Ratu Textile*.

Rabbama. 2009. *Programmable Logic Controller* (Online). <http://www.scribd.com/doc/22163532/PLC-Programmable-Logic-Controller>, diakses tanggal 19 Januari 2010.

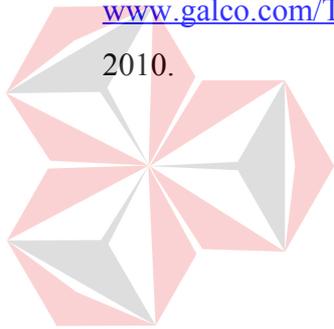
STIKOM Surabaya. 2008. *Kilas Sejarah* (Online). <http://stikom.edu>, diakses tanggal 21 Januari 2010.

STIKOM Surabaya. 2008. *Visi & Misi* (Online). <http://stikom.edu>, diakses tanggal 21 Januari 2010.

Teknik Instalasi Pemanfaatan Tenaga Listrik. 2009. *Dasar-dasar PLC* (Online). <http://dunia-listrik.blogspot.com>, diakses tanggal 19 Januari 2010.

Toshiba Corporation. 2003. *Instruction Manual TOSVERT TM VF-S11*. Japan.

Toshiba Corporation. 2004. *TOSVERT VF-S11 Instruction Manual* (Online). [www.galco.com/Techdoc/TOSH/S11/S11\\_SER.pdf](http://www.galco.com/Techdoc/TOSH/S11/S11_SER.pdf), diakses tanggal 20 Januari 2010.



UNIVERSITAS  
**Dinamika**