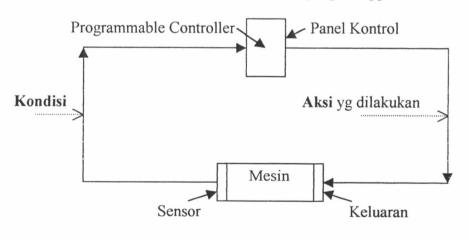
## **BABII**

# **TEORI PENUNJANG**

# 2.1 PLC (PROGRAMMABLE LOGIC CONTROL)

### 2.1.1 Pendahuluan

PLC (Programmable Logic Control) dewasa ini sangatlah dibutuhkan, hal ini dikarenakan pemakaiannya yang sangat praktis dan fleksibilitas. PLC dapat mengendalikan atau menggerakkan suatu peralatan atau komponen tergantung pada program yang diinginkan. Untuk dapat bekerja PLC membutuhkan masukan yang menginformasikan kondisi mesin atau peralatan melalui rangkaian sensor yang kemudian diolah dan diproses oleh rangkaian dalam PLC, lalu disesuaikan dengan kondisi-kondisi yang diinginkan oleh program sehingga menghasilkan keluaran untuk menggerakkan peralatan atau sebagai aksi dari program. Agar lebih jelasnya lihatlah gambar sistem control yang menggunakan PLC.



Gambar 2.1

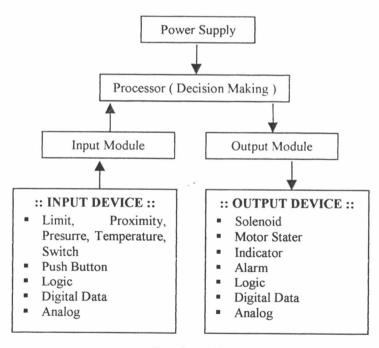
Dasar Sistem Kontrol PLC

Suatu program menginformasikan programmable controller tentang apa yang harus dikerjakan. Suatu program tidak akan ada yang melakukan setting lebih dari instruksi yang diberikan pada programmable controller yang memberitahukan bagaimana melakukan aksi untuk kondisi tertentu pada mesin.

### 2.1.2 Teori Dasar PLC

Sistem Programmable controller yang biasa terdiri atau meliputi 4 bagian yaitu :

- 1. Processor.
  - 2. Masukan (Input Module).
  - 3. Keluaran (Output Module).
  - 4. Catu Daya (Power Supply).



Gambar 2.2

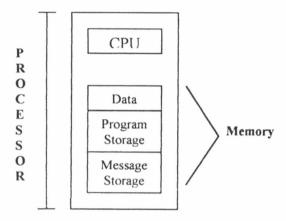
Sistem Bagian dari PLC

Adapun keterangan selanjutnya ke empat bagian diatas akan dijelaskan seperti bawah ini.

#### 2.1.2.1 Processor

Processor dapat juga dinamakan "OTAK" dari programmable controller. Processor dibagi menjadi dua bagian, yaitu :

- a. CPU (Central Processing Unit)
- b. Memory



Gambar 2.3

Bagian dari processor

## A. CPU (Central Processing Unit)

Central processing unit (CPU) melakukan suatu keputusan atas tindakan yang harus dilakukan processor sesuai dengan program yang ditulis. Semua data yang yang masukyng berupa informasi tentang kondisi suatu peralatan akan diproses dalam unit ini.

## B. Memory

Memory disini ada tiga fungsi, yaitu:

- a. menyimpan informasi ke dalam data table yang mana CPU memerlukannya untuk menjalankan program.
- b. Menyimpan program.
- c. Menyimpan pesan.

### **B.1.** Data Table

Data disimpan dan diuji didalam suatu area memory yang disebut Data Table. Data table dibagi menjadi beberapa bagian yang lebih kecil sesuai dengan jenis informasi yang disimpan.

Bagian yang lebih kecil itu adalah:

- a. input image table
- b. output image table
- c. timer / counter storage

Data Table

Output Image Table

Input Image Table

Timer / Counter Storage

Gambar 2.4

Bagian dari Data Table

### **Image Table**

Input image table menyatakan status atau keadaan dari terminal terminal module yang dikendalikan oleh program. Masing-masing image table dibagi dalam bagian-bagian yang lebih kecil yang disebut bit. Bit adalah bagian terkecil dari memory. Ketika processor mendeteksi tegangan pada terminal masukan, processor menulis atau mencatat informasi itu, dengan menghidupkan atau mengaktifkan bit-bit yang sesuai didalam input image table. Begitu juga bila processor tidak mendeteksi tegangan pada terminal masukan, processor menulis informasi itu dengan tidak mengaktifkan bit yang sesuai.

#### a. Output Image Table

Tujuan dari *Output Image Table* adalah mengontrol kondisi ON/OFF. Suatu peralatan keluaran yang dihubungkan ke terminal-terminal output module. Bila sebuah bit *output image table* ON (1), maka peralatan keluaran yang sesuai dengan *output image table* akan ON (diaktifkan atau berenergi), begitu juga bila sebuah bit *output image table* OFF (0), maka peralatan keluaran yang sesuai dengan *output image table* akan OFF (tidak aktif atau tidak berenergi). Bit-bit *output image table* dikendalikan oleh pemakai program.

### b. Input Image Table

Tujuan dari *input image table* adalah mencatat kondisi ON atau OFF dari peralatan masukan. Bila peralatan masukan ON (close), maka peralatan masukan yang sesuai dengan *input image table* akan ON (1), jika sebuah

input image table OFF (open), maka peralatan masukan yang sesuai dengan input image table akan OFF (0). Bit-bit input image table dimonitor oleh pemakai program.

### c. Timer / Counter Storage

Instruksi timer dan counter adalah instruksi-instruksi keluaran. Keduanya mempunyai informasi yang telah disimpan pada data table. Ada beberapa instruksi yang memerintahkan processor untuk membaca apa-apa yang ditulis pada image table. Ketika processor diperintahkan untuk menulis kedalam *output image table*, processor menulis 1 atau 0 ke dalam bit-bit khusus. Sebagai hasilnya peralatan keluaran akan ON/OFF, banyak kemampuan yang dapat menggerakkan relay-relay pewaktu, pewaktu solid state dan peralatan penghitung. Biasanya menentukan suatu kondisi maka akan ditentukan oleh instruksi pengetesan. Keduanya menjaga interval waktu atau menghitung kejadian-kejadian sesuai dengan logikanya.

### **B.2.** Program Storage

Program storage mempunyai porsi yang besar dari memory, dimana tempat pemprogram menyimpan programnya. Adapun programnya diset dari statement, dan setiap statement mempunyai dua hal antara lain:

- Menggambarkan kondisi bahwa harus ada pesan atau instruksi untuk dapat melakukan aksi.
- Menggambarkan aksi yang diinginkan misalnya : "aktifkan motor nomor satu".

Statement	
Statement	

Gambar 2.5

# Program Storage Area Memory

Jika menginginkan aksi untuk mengaktifkan motor satu, maka limit switch harus tertutup. Tiap-tiap statemen mencatat kondisi-kondisi yang harus bertemu, dan kemudian melakukan aksi. Tiap-tiap kondisi dinyatakan oleh instruksi tertentu, dimana instruksi-instruksi memberitahukan processor untuk mengerjakan sesuatu dengan informasi yang telah disimpan pada data table. Ada beberapa instruksi yang memerintahkan processor untuk membaca, apa-apa yang ditulis pada image table. Ketika processor diperintahkan untuk membaca *input image table*, processor tersebut menguji bit khusus untuk melihat bila peralatan masukan tertentu itu ON atau OFF. Instruksi lain memberitahukan untuk menulis kedalam output image table processor menulis 1 atau 0 kedalam bit-bit khusus. Sebagai hasilnya peralatan keluaran akan ON atau OFF.

## **B.3.** Message Storage

Area memory yang ketiga adalah message storage, dimulai setelah statement pemakaian program. Dua alphanumeric karakter dapat disimpan dalam kata. Pesan dimasukkan ke dalam memory dari terminal 1770 – T3 peralatan peripheral. Pesan ditampilkan pada terminal 1770 – T3 atau peralatan peripheral setiap waktu pesan itu diinginkan. Pesan-pesan diaktifkan melalui kontrol program oleh instruksi khusus pemprograman pada program diagram ladder.

### 2.1.2.2 Modul Masukan (Input Module)

Modul masukan (input modul) dari programmable controller mempunyai 4 fungsi yaitu :

- a. Terminal.
- b. Indikator.
- c. Penyesuaian atau sinkronisasi.
- d. Isolasi atau pelindung.

Keterangannya adalah sebagai berikut :

#### a. Terminal

Modul masukan (input module) tersedia beberapa terminal untuk menghubungkan kabel dari rangkaian sensor pada suatu mesin.

#### b. Indikator

Modul masukan (input module) yang sebagian besar terdiri dari indikator secara visual (misalnya LED indikator) yang menunjukkan keadaan dari masingmasing terminal masukan. Indikator ON bila pada terminal itu ada tegangan. Indikator OFF bila pada terminal indikator tidak ada tegangan. Ketika indikator menunjukkan keadaan dari terminal, hal ini sering disebut *input status indicator*.

Indikator masukan hanya dihubungkan dengan terminal yang digunakan untuk kabel sensor masukan (input module). Terminal untuk ground pada rangkaian sensor tidak ada indikatornya.

## c. Penyesuaian atau sinkronisasi

Fungsi lain dari modul masukan (input module) adalah untuk menyesuaikan sinyal. Level tegangan yang digunakan pada mesin sering tidak sama atau tidak sesuai dengan level tegangan pada programmable controller. Modul masukan (input module) menerima sinyal listrik dari mesin dan sinyal itu diubah ke level tegangan yang sesuai dengan rangkaian pada programmable controller.

## d. Isolasi atau Pelindung

Modul masukan (input module) memisahkan rangkaian mesin dari rangkaian programmable controller. Pemisahan ini membantu melindungi rangkaian programmable controller dari bahaya level tegangan yang tidak diinginkan.

## 2.1.2.3 Modul Keluaran (Output Module)

Modul keluaran (output module) pada programmable controller mempunyai 4 fungsi yaitu :

- a. Terminal.
- b. Indikator.
- c. Penyesuaian atau sinkronisasi.
- d. Isolasi atau pelindung.

Keterangannya adalah sebagai berikut:

#### a. Terminal

Modul keluaran (output module) mempunyai terminal untuk menghubungkan kabel ke rangkaian keluaran pada mesin.

#### b. Indikator

Modul keluaran (output module) sebagian besar terdiri dari indikator secara visual (misalnya LED) yang menunjukkan keadaan dari masing-masing terminal output.

Dalam modul yang berkapasitas rendah, bila power atau daya ada pada terminal output, maka indikator akan ON. Sedangkan modul berkapasitas tinggi, daya atau power tidak mungkin ada pada terminal keluaran untuk kondisi indikator ON.

## c. Penyesuaian atau sinkronisasi

Modul keluaran (output module) programmable controller menghasilkan sinyal-sinyal listrik yang bisa dipakai oleh rangkaian keluaran yang dihubungkan ke terminal.

#### d. Isolasi atau Pelindung

Modul keluaran (output module) melindungi rangkaian programmable controller dari tegangan-tegangan yang membahayakan dan tegangan-tegangan yang tidak diinginkan pada mesin. Beberapa keadaan memerlukan tambahan proteksi untuk melindungi secara eksternal.

## 2.1.2.4 Catu Daya (Power Supply)

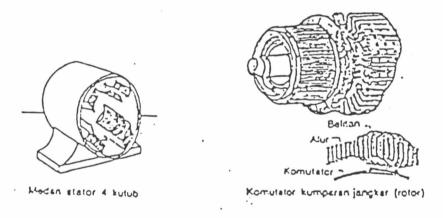
Catu daya (Power Supply) menghasilkan tegangan searah (DC) untuk rangkaian elektronik dari processor, input dan output module. Power Supply mengubah sumber tegangan dari tegangan logika yang lebih rendah yang diperlukan oleh processor, *input dan output module*.

### 2.2 MOTOR DC

Motor DC (*direct current*) adalah sebuah motor yang digerakkan oleh arus listrik searah atau DC. Pada motor DC terdapat dua macam jenis dilihat dari cara pemberian catu daya yaitu motor DC yaitu motor DC reverseable yaitu motor DC yang catu dayanya bisa dibolak-balik posisinya sehingga motor dapat berputar ke kanan atau ke kiri dan motor DC non reverseable yaitu motor DC

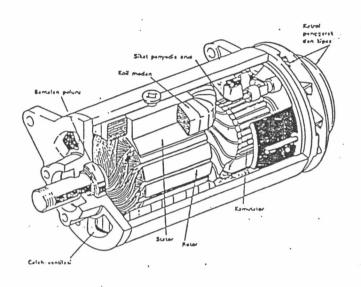
yang catu dayanya tidak bisa dibolak-balik posisinya sehingga motor hanya berputar pada satu arah saja.

Penampang motor Desecara umum yang ada dipasaran ditunjukkan pada gambar 2.6 penampang motor de. Pada motor terdapat dua bagian utama yaitu bagian yang tidak berputar yang disebut dengan stator dan bagian yang berputar yang disebut rotor.



Gambar 2.6 Penampang Motor DC

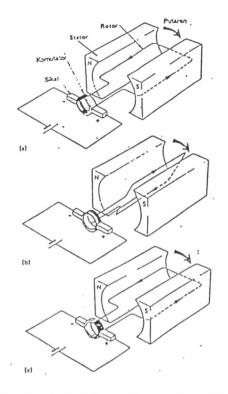
Pada motor DC bagian stator berupa kumparan medan yang berbentuk kutup sepatu atau magnet permanen yang menghasilkan medan magnet, bagian rotor berupa kumparan jangkar yang dialiri arus listrik. Bagian – bagian motor DC diperlihatkan dalam gambar 2.7



Gambar 2.7 Bagian – bagian motor DC

# 2.2.1 Prinsip Kerja Motor DC

Prinsip kerja motor DC dapat dilihat pada gambar 2.8. Arus listrik mengalir pada kumparan melalui sikat – sikat yang yang selalu berhubungan dengan komutator yang ditekan oleh pegas. Pada posisi seperti gambar 2.8 (a), aliran arus pada kumparan akan menghasilkan medan magnet yang berlawanan dengan medan magnet dari magnet stator, hal ini menyebabkan kumparan berputar kearah yang ditunjukkan oleh anak panah.



Gambar 2.8 Prinsip Kerja Motor DC

Apabila aliran arus tetap mengalir dalam arah seperti gambar 2.8 (a), kumparan akan diam pada posisi vertikal setelah berputar sejauh 90. Untuk lebih lanjut kita harus memahami tujuan dari kumutator. Karena adanya momentum dari kumparan, maka gerakannya akan mnelewati posisi vertikal. Apabila kumparan telah mencapai posisi seperti yang tampak pada gambar 2.8 (b), komutator akan menyebabkan aliran arus yang mengalir melalui kumparan berbalik rarh dari arah selmula. Dengan demikian aliran arus yang terdapat pada kumparan pada posisi ini menghasilkan tolakan magnet yang memutar kumparan sejauh 90 keposisi seperti gambar 2.8 (c). Kemudian proses ini akan kembali pada proses awal dan hal ini terus terjadi berulang – ulang menyebabkan motor berputar.

#### 2.3 RELAY

Relay pada dasarnya sakelar elektro magnet. Relay sederhana yang mempunyai satu kontak dan dengan sistem kumparan sederhana. Kalau ada arus yang mengalir didalam kumparan, inti besi menjadi magnet. Maka jangkar yang terbuat dari besi lunak tertarik, dan bergerak mengguling pada engsel, karena itu relay semacam ini dinamakan *relay guling*.

Relay dapat berguling, kalau gaya magnet dapat mengalahkan gaya pegas yang melawannya, maka kontak pun menutup. Besar gaya magnet ditetapkan oleh kuat medan magnet yang ada didalam celah udara, diantara jangkar dan inti. Adapun kuat medan magnet ini tergantung pula pada banyaknya lilitan kumparan dan kuat arus yang ada pada kumparan itu, atau dengan singkatnya bergantung pada ampere lilitan.

Relay ada banyak sekali variasinya, namun bekerjanya berdasarkan pada asas tersebut diatas juga. Banyaknya kontak, besar gaya yang diperlukan dan cara bagaimanakah gaya itu dilimpahkan kepada kontak-kontak itu semua bergantung kepada tujuan penggunaan relay.

Relay akan dapat dipakai untuk beberapa jenis keperluan.

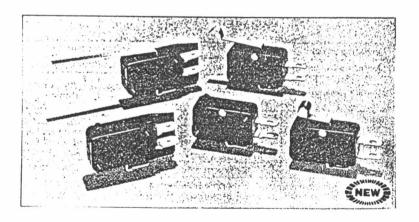
- 1. Relay bekerja sebagai saklar
- 2. Relay bekerja sebagai penguat
- 3. Relay dapat menyimpan informasi

Jika hendak memiliki relay untuk suatu keperluan, maka yang penting untuk diperhatikan adalah apakah kontak — kontak mampu untuk diberi beban

sebesar yang dikehendaki. Arus yang terlampau besar yang mengalir pada kontak, akan dapat merusak kontak itu sendiri.

## 2.4 MICROSWITCH

Microswitch merupakan saklar mikro yang dapat memutus ataupun menghubungkan suatu aliran listrik. Sebenarnya microswich tidak berbeda jauh dengan saklar on/off yang biasa, namun kelebihan dari microswitch ini hanya dengan sedikit sentuh atau gerakan ringan alat ini akan berubah posisi. Hal tersebut memungkinkan microswitch difungsikan sebagai sensor pengaman (safety) atau sensor – sensor lain pada suatu peralatan tertentu. Berbagai macam bentuk dan ukuran miroswitch ini bervariasi mulai dari yang kecil hingga yang besar dapat dipilih sesuai dengan kebutuhan.



Gambar 2.9 Micro Switch

# 2.5 FST (FESTO SOFTWARE TOOLS) PROGRAMMING

Festo software tools (FST) diciptakan untuk membantu kinerja semua orang dalam membuat programming logic controller. FST dapat dioperasikan pada IBM XT/AT komputer kompatibel menggunakan PC/MC-DOS dan dapat digunakan untuk beberapa bahasa program diantaranya :

- Matrix
- Statement List
- Ladder Diagram
- BASIC

Pada penulisan tugas akhir ini penulis menggunakan Statement List Language (STL) dalam pembuatan program lift.

# 2.5.1 Operand

Seperti halnya dengan bahasa program yang lainnya pada FST software juga memiliki beberapa operan. Dalam penulisan program FST membolehkan penulisan operan secara absoluth ( *absoluth operands* ) yakni penulisan yang sesuai dengan bagian yang sebenarnya ( misal : T1 adalah operan absoluth untuk timer nomer 1 ). Dan Simbolik operan ( *Symbolic Operands* ) yaitu penulisan dengan menggunakan simbol ( misal O.0.1 untuk motor ). Ada beberapa operan dalam FST diantaranya :

## Single dan Multi bit operand

Single bit operan (SBO) dapat dievaluasi / koreksi sebagai benar/salah pada bagian kondisi dalam kalimat program dan dapat Set / Reset pada bagian khusus dalam kalimat program . Single bit operan diletakkan dalam Single Bit Accumulator (SBA) pada CPU

Multibit operan dapat diuji untuk nilai (<, >, =, dll) atau dibandingkan untuk Multibit operan lain . Multibit operan dapat digunakan untuk pengurangan , penambahan atau manipulasi melalui arithmetic logic (ALU).

## Lokal dan Global Operand

Lokal Operan adalah bagian dari sistem yang dapat diakses oleh beberapa program didalam bagian tunggal CPU. Yang termasuk dalam Operan Lokal diantaranya: Timer, Counters, Registers.

Global Operan adalah bagian sistem yang dapat diakses oleh program didalam CPU. Beberapa contoh global operan diantaranya : Input , Output , Flags

### Berikut ini adalah tabel dari SBO:

Operand	STL Form	Syntax	Part	Typical Exp.
Input	I	In.n	С	IF I2.0
Output	0	On.n	С	IF O2.6
Output	О	On.n	E	SET O2.3
Flags	F	Fn.n	С	IF F1.16
Flags	F	Fn.n	Е	RESET F9.3
Counter	С	Cn	С	IF C3

Counter	С	Cn	Е	SET C5
Timer	Т	Tn	С	IF T7
Timer	T	Tn	Е	SET T4
Program	Р	Pn	С	* IF P2
Program	P	Pn	Е	* SET P3
Processor	Y	Yn	С	* IF Y2
Processor	Y	Yn	Е	* RESET Y1
Error Status	Е	Е	С	* IF E
Auto Restart	ARU	ARU	С	* IF ARU

<sup>\* =</sup> operands yang terdapat tanda bintang tidak dapat digunakan diseluruh model controller.

# Berikut ini adalah tabel dari MBO:

Operand	STL Form	Syntax	Part	Typical Exp.
Input Word	IW	Iwn	С	IF (IW3 = V255)
Output Word	OW	FWn	С	IF (OW2 = V80)
Output Word	OW	FWn	Е	Load V128 to OW3
Flags Word	FW	FWn	С	IF (FW3 = V220)
Flags Word	FW	FWn	Е	Load V2 1000 to FW1
Function Unit	FU	Fun	С	IF (FU32 = V16)
Function Unit	FU	Fun	E	Load FU34 to R60
Timer Word	TW	TWn	С	IF (TW2 < V2000)
Timer Word	TW	TWn	Е	Load V1345 to TW6
Timer Preselect	TP	TPn	С	IF (TP0 < V20)
Timer Preselect	TP	TPn	E	* Then Load V500 to TP4
Counter Word	CW	CWn	С	IF (CW3 <> V50)
Counter Word	CW	CWn	Е	Then INC CW5

Counter Preselect	СР	CPn	С	IF (CP3 = V555)
Counter Preselect	СР	CPn	E	Load V67 to CP5
Register	R	Rn	С	IF (R60 = V21113)
Register	R	Rn	Е	Load (R53 + R45) to R32
Error Word	EW	EW	С	IF (EW and V15)
Error Word	EW	EW	Е	Load V0 to EW

<sup>\* =</sup> operands yang terdapat tanda bintang tidak dapat digunakan diseluruh model controller.

## 2.5.2 Operator

Berikut ini merupakan tabel dari operator Statement List:

Symbol	Tujuan
N	Not (negation)
V	Nilai untuk operand desimal
V\$	Nilai untuk operand hexadesimal
V%	Nilai untuk operand biner
+	Penambahan
-	Pengurangan
*	Perkalian
/	Pembagian
<	Lebih kecil
>	Lebih besar
	Sama dengan
$\Diamond$	Tidak sama dengan
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan
()	Kurung buka / kurung tutup digunakan untuk spesifikasi operasi arithmatika

# 2.5.3 Struktur Program STL (Statement List)

Statement list program dibangun dengan menggunakan beberapa element penting. Dibawah ini merupakan beberapa element yang digunakan dalam sebuah program STL :

- Program
- Step

- Sentence
- Conditional Part
- Execution Part

### Program

Program yang dimaksud adalah kepala / judul program, yang berguna untuk memberikan penjelasan tentang maksud dari program.

## Step

Instruksi dalam program STL yang berupa langkah-langkah program. Program yang menggunakan atau tidak menggunakan step dapat diprogram. STL menyediakan instruksi step dimaksudkan agar program dapat dibagi menjadi beberapa bagian yang dapat berdiri sendiri.

Contoh Program menggunakan step:

Step (label)

IF I1.0 (Jika Input I1.0 aktif)

THEN SET O2.4 (maka aktifkan output O2.4 dan proses ke

langkah berikutnya)

Contoh Program tidak menggunakan step:

IF I1.0 (Jika Input I1.0 aktif)

THEN SET O1.2 (maka aktifkan output O2.4)

### **Sentence**

Sentence merupakan isi program.

### **Conditional Part**

Conditional part merupakan statement program yang menandakan sebuah kondisi dimana pada akhir statement bernilai "true/false" atau "set/reset".

Conditional part diawali dengan "IF" dan disambung dengan "THEN".

### **Execution Part**

Execution part merupakan statement program yang berfungsi sebagai eksekusi dari conditional part.

#### 2.5.4 Instruksi – Instruksi

Dibawah ini merupakan beberapa instruksi dari Statement List:

- AND → Sebuah instruksi yang menyatakan kondisi 2 operand akan set (bernilai=1) jika kedua operand itu bernilai 1 dan 1. Dan sebaliknya jika kedua operand bernilai 1 dan 0, 0 dan 1, 0 dan 0 maka akan reset (bernilai=0).
- OR → Sebuah instruksi yang menyatakan kondisi 2 operand akan set (bernilai=1) jika kedua operand itu bernilai 1 dan 0, 0 dan 1, 1 dan 1. Dan sebaliknya jika kedua operand bernilai 0 dan 0, maka akan reset (bernilai=0).
- IF 
   → Sebuah instruksi yang menyatakan tahap awal dari sebuah kondisi dari sebuah kalimat.
- JMPTO → Sebuah instruksi yang menyatakan perintah untuk menuju pada langkah yang ditunjuk.

- SET → Sebuah instruksi yang berarti memberikan nilai 1 pada sebuah operand.
- RESET → Sebuah instruksi yang berarti memberikan nilai 0 pada sebuah operand
- OTHRWS → Sebuah instruksi yang akan dilakukan jika eksekusi program bernilai false.

# 2.5.5 Timer, Counter, Register dan Flag

### 2.5.5.1 Timer

Timer merupakan operand yang digunakan untuk menyimpan nilai setting dan nilai aktual pada sebuah statement program.

### 2.5.5.2 Counter

Counter merupakan operand yang digunakan untuk penghitungan kembali dalam sebuah ststement

## 2.5.5.3 Register

Register merupakan operand yang digunakan untuk accumulator.

## 2.5.5.4 Flag

Flag berfungsi sebagai pengingat kondisi program.