

LAMPIRAN

1. PLC Allen Bradley

PLC merupakan singkatan dari *Programmable Logic Controller*. Alat ini merupakan salah satu jenis dari *microcontroller* dan merupakan sistem elektronik yang beroperasi secara digital dan didesain untuk pemakaian di lingkungan industri, dimana sistem ini menggunakan memori yang dapat diprogram untuk penyimpanan secara internal instruksi-instruksi yang mengimplementasikan fungsi-fungsi spesifik seperti logika, urutan, pewaktuan, pencacahan, dan operasi aritmatik untuk mengontrol mesin atau proses melalui modul-modul I/O *digital* maupun analog.

Berdasarkan namanya, konsep PLC adalah sebagai berikut:

- a. Programmable, menunjukkan kemampuan dalam hal memori untuk menyimpan program yang telah dibuat yang dengan mudah diubah-ubah fungsi atau kegunaannya.
- b. Logic, menunjukkan kemampuan dalam memproses input secara aritmatik dan logika (ALU = Arithmetic Logic Unit), yakni melakukan operasi perbandingan, penjumlahan, perkalian, pembagian, pengurangan, negasi, AND, OR, dan lain sebagainya.
- c. Controller, menunjukkan kemampuan dalam mengontrol dan mengatur proses sehingga menghasilkan output yang diinginkan.

Dalam hal ini PLC yang digunakan adalah PLC Allen Bradley. PLC ini merupakan salah satu jenis PLC yang banyak digunakan untuk keperluan otomasi di industri. Ada dua macam bentuk PLC Allen Bradley yaitu bentuk *compact* dan bentuk *modular*. Untuk bentuk *compact*, PLC-nya menggunakan sistem rak (CPU dan I/O jadi satu kesatuan) dengan kapasitas memori yang terbatas.



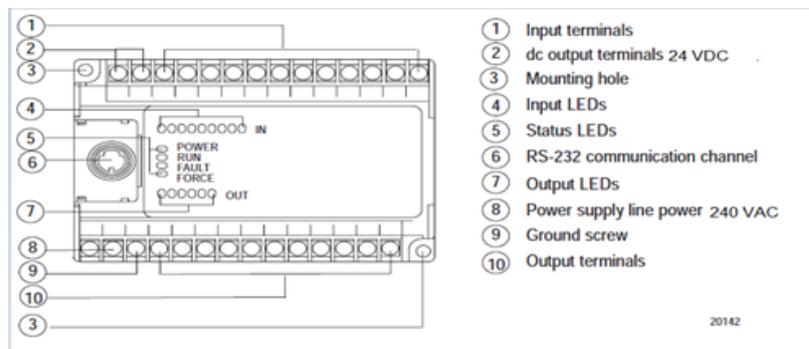
Gambar 1 Bentuk PLC Compact

Sedangkan untuk PLC bentuk *modular* terdiri dari modul CPU dan I/O (merupakan bagian yang terpisah-pisah).



Gambar 2 Bentuk PLC Modular

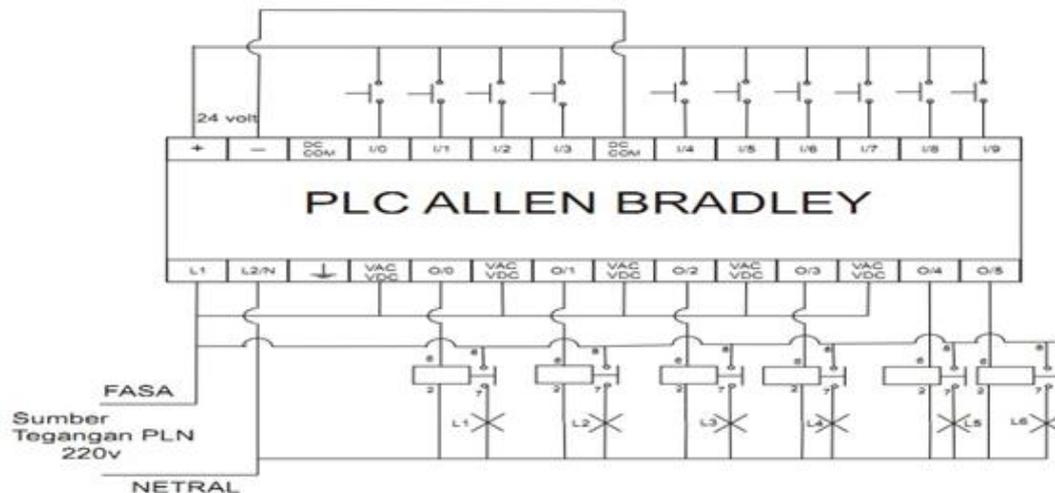
Untuk bentuk PLC yang digunakan adalah bentuk *compact* dengan tipe PLC MikroLogix 1000 yang memiliki I/O berjumlah 10/6. Berikut gambar dari PLC beserta keterangannya.



Gambar 3 Konfigurasi PLC Allen Bradley

a. Pemasangan Instalasi PLC AB

Konstruksi pemasangan instalasi PLC secara umum sama, yang membedakan adalah peletakan sumber tegangan PLN pada PLC (VAC) dengan output keluaran PLC (VDC). Untuk instalasi input dan output PLC Allen Bradley dijelaskan seperti Gambar 1.4 berikut:



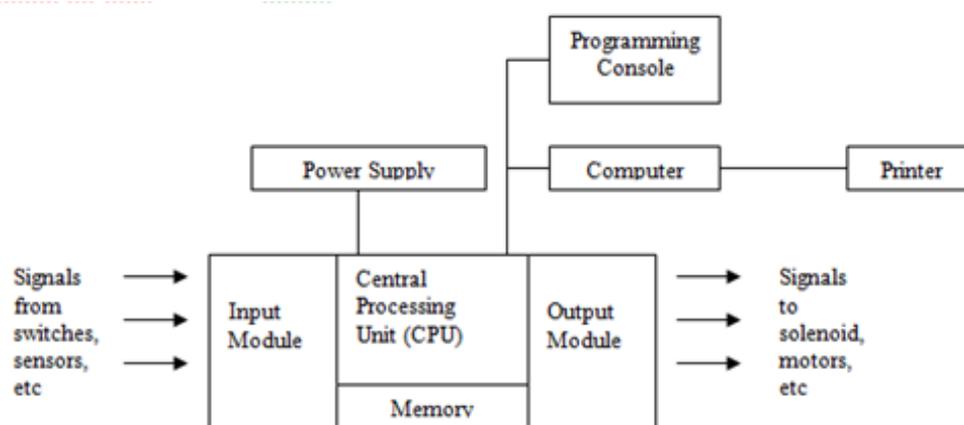
Gambar 4 Instalasi PLC Allen Bradley

Dari konstruksi diatas pemasangan instalasi PLC, rangkaian input dapat berupa saklar toggle, saklar tekan, sensor proximity, dan sebagainya. Tegangan input dapat diambil dari tegangan VDC yang merupakan tegangan keluaran dari PLC. Sedangkan rangkaian output dapat berupa lampu indikator, solenoid, timer, motor. Untuk tegangan output dapat diperoleh dari sumber tegangan PLC yaitu VDC, yang merupakan tegangan masukan dari PLN.

b. Arsitektur PLC AB

➤ Komponen PLC

Secara umum PLC memiliki bagian-bagian yang sama dengan komputer maupun mikrokontroler, yaitu CPU, Memori dan I/O. Susunan komponen PLC dapat dilihat pada gambar berikut



Gambar 5 Bagian-bagian PLC

➤ Unit Pengolah Pusat (CPU - Central Processing Unit)

CPU merupakan bagian utama dan merupakan perangkat utama dari sebuah PLC. CPU ini berfungsi untuk mengambil instruksi dari memori kemudian mengkodekan dan mengeksekusi instruksi tersebut, CPU menghasilkan sinyal kendali sesuai dengan proses program yang ada, menghubungkan input dan output sesuai instruksi yang digunakan. Pada dasarnya CPU terdiri dari atas register, *Control Unit* (CU), dan *arithmetic Logikal Unit* (ALU). Register merupakan penyimpanan data sementara yang dapat digunakan selama pengekseskuan program.

➤ Memori

Memori merupakan bagian CPU yang berfungsi sebagai penyimpanan program. Menurut jenisnya memori dapat dibagi menjadi dua macam yaitu: (1) ROM (*Random Only Memory*), adalah elemen memori yang ditempatkan dalam sebuah chip rangkaian terpadu yang isinya tidak dapat diubah oleh programmer. ROM berisi suatu pola yang tetap dari data-data biner yang telah dibentuk pada saat ROM tersebut dibuat. ROM menyimpan data-data penting untuk operasi PLC dan data tersebut tidak akan hilang apabila catu daya dimatikan. (2) RAM (*Random Acces Memory*), merupakan penyimpanan data yang digunakan sesaat dalam operasi program dan data dapat dituliskan ke dalam *adres* atau alamat pada *image table*.

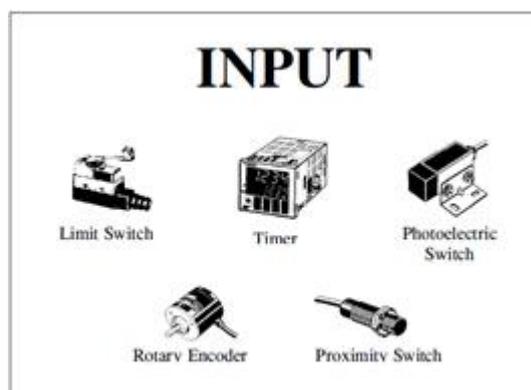
➤ **Catu daya PLC**

Catu daya atau *power supply* pada PLC mempunyai fungsi sebagai pemberi tegangan listrik kepada seluruh bagian PLC (termasuk CPU, memori dan lain-lain). Catu daya PLC sendiri kebanyakan bekerja pada tegangan catu daya sekitar 24 VDC / 220 VAC. Selain berfungsi sebagai penyedia tegangan listrik, catu daya juga dapat berfungsi sebagai pemonitor dan memberikan sinyal kepada CPU apabila terjadi suatu kesalahan.

➤ **Peralatan Input**

Peralatan input adalah peralatan yang memberikan sinyal masukan kepada PLC dan selanjutnya PLC memproses sinyal masukan tersebut untuk mengendalikan peralatan output yang digunakan. Peralatan input PLC yang sering digunakan antara lain:

- Berbagai jenis saklar, misalnya saklar togel, saklar tekan, saklar proximity.
- Berbagai jenis sensor, misalnya sensor cahaya, sensor suhu, sensor level.
- Rotary encoder

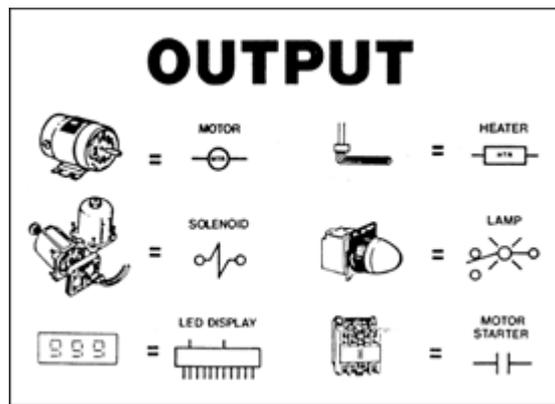


Gambar 6 Peralatan Input

➤ **Peralatan Output**

Pada dasar peralatan output pada PLC dipengaruhi dua hal yakni peralatan output internal dan peralatan output eksternal. Peralatan output internal adalah peralatan yang berupa relay internal, timer internal, dan lain-lain yang ada pada

PLC. Sedangkan peralatan output eksternal adalah peralatan output yang merupakan suatu keluaran dari PLC yang dikendalikan. Seperti gambar dibawah ini beberapa contoh peralatan output yang umum digunakan.



Gambar 7 Peralatan Output

c. Fungsi dan Prinsip Kerja PLC

Fungsi dan kegunaan PLC sangat luas. Dalam prakteknya PLC dapat dibagi secara umum dan secara khusus. Secara umum fungsi PLC adalah sebagai berikut: 1) *Sequential Control*. PLC memproses input sinyal biner menjadi output yang digunakan untuk keperluan pemrosesan teknik secara berurutan (sekuensial), disini PLC menjaga agar semua step atau langkah dalam proses sekuensial berlangsung dalam urutan yang tepat. 2) *Monitoring Plant*. PLC secara terus menerus memonitor status 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 pada operator.

d. Instruksi Dasar PLC AB

Instruksi dasar di dalam pembuatan *diagram ladder* pada PLC AB secara keseluruhan menggunakan instruksi bit. Instruksi yang digunakan untuk memprogram masih sederhana, instruksi tersebut diantaranya:

- Examine if Closed (XIC)



Fungsi: Menentukan status bit B sebagai kondisi eksekusi untuk operasi selanjutnya di dalam suatu baris instruksi.

- Examine if Open (XIO)



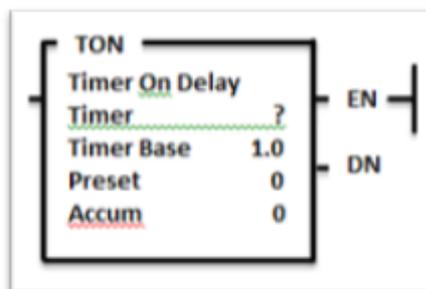
Fungsi: Menentukan status dari invers bit B sebagai kondisi eksekusi untuk operasi selanjutnya didalam suatu baris intruksi.

- Output Energize (OTE)



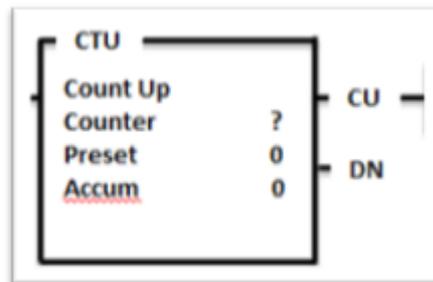
Fungsi: Status bit B *ON* untuk suatu kondisi eksekusi *ON* dan status bit B akan *OFF* untuk sutu kondisi eksekusi *OFF*.

- Pewaktu (Timer)



Fungsi: Timer pada jenis ini terdiri dari Timer On Delay (TON) dan Timer OFF Delay (TOF). Alamat pada timer dimulai T4:00 sampai 39. Timer ini dilengkapi dengan bit yang terdiri dari EN (Timer Enable Bit), TT (Timer Timing Bit), dan DN (Timer Done Bit).

➤ Pencacah (Counter)



Counter terdiri dari 2 bagian yaitu CTU (Counter UP) dan CTD (Counter Down). Dalam instruksi counter terdapat *Preset* yang berfungsi untuk hitungan dimana DN akan aktif dan *Accum* yang berfungsi sebagai nilai counter. ketika *Accum* sama dengan *Preset*, DN akan aktif

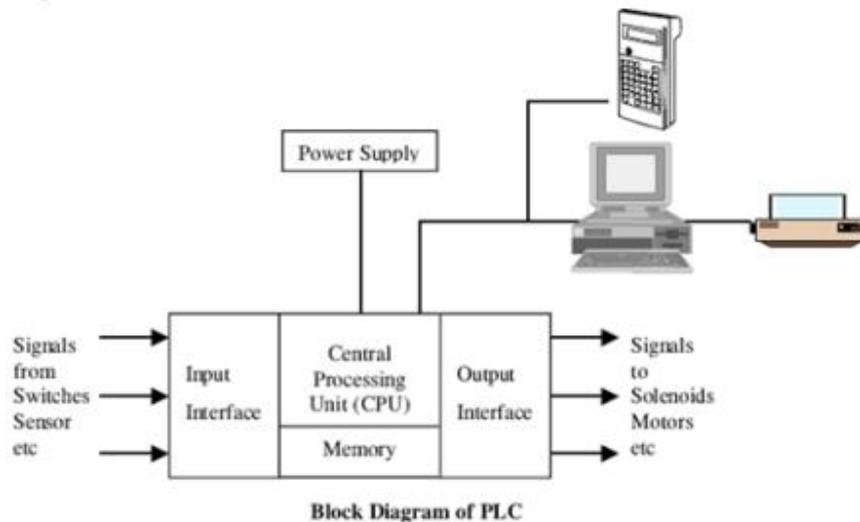
➤ Reset (RES)



Fungsi: untuk mereset nilai accum dari suatu counter/ timer hingga nilai menjadi nol.

2. PLC SIEMENS

Programmable logic controller (PLC) adalah suatu system control yang terdiri dari peralatan elektronik yang digunakan untuk mengontrol proses tertentu. PLC merupakan bagian system control yang terhubung dengan komponen control seperti switch, solenoid, dan sensor-sensor lainnya. PLC yang merupakan bagian control system terhubung langsung dengan proses control yang dilakukan sesuai dengan program yang ada pada memori. PLC akan menerima data sinyal masukan, memprosesnya dan menghasilkan keluaran sebagai data keluarannya.



Gambar 8 Block diagram

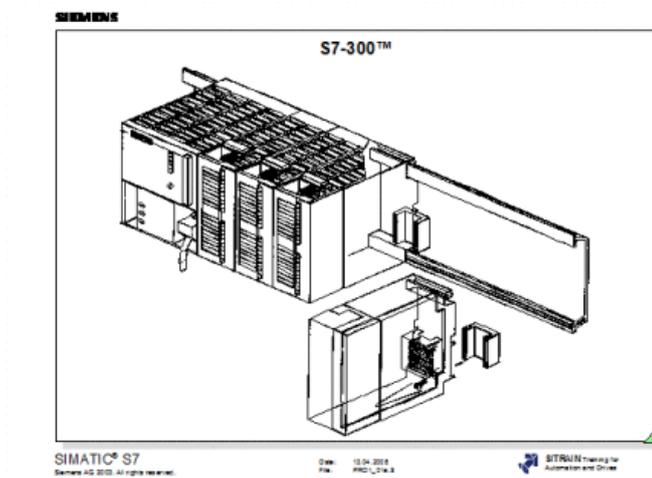
Dari gambar blok diagram diatas dapat dijelaskan sebagai berikut:

- Input interface (masukan) : berupa sinyal dari peralatan lain seperti photoelektrik, microswitch, pushbutton, dan masih banyak lagi, baik dalam bentuk analog maupun digital yang digunakan sebagai signal instruksi dalam program.
- Output interface (keluaran) : hasil dari eksekusi suatu program yang terhubung dengan peralatan lain seperti relay, lampu indicator dan lain-lainnya. Output ini bisa berupa analog maupun digital dan AC ataupun DC.
- Power Supply (catu daya): merupakan unit yang menyediakan daya untuk kebutuhan PLC.
- Central Processing Unit (CPU): suatu perangkat yang mampu untuk menyimpan data dan program. Didalam PLC digunakan untuk memproses program, sinyal I/O serta berkomunikasi dengan peralatan lain.
- Memory: tempat menyimpan program dan data aplikasi yang ada pada PLC.

Simatic Controller sebagai PLC conventional Siemens dapat dihubungkan ke semua jaringan via integrated interface atau communication processor yang meliputi sistem dibawah ini:

- Industrial Ethernet (IEEE 802.3 dan 802.3u)

- PROFINET
- PROFIBUS (IEC 61158/EN 50170)
- AS-Interface
- KNX (EN 50090, ANSI EIA 776)
- Point-to-Point Connection (PPI), dengan special protocol RK 512, 3964(R), dan ASCII
- Multipoint-enabled Interface (MPI)



Gambar 9 Modul PLC Siemens S7-300

Berikut ini adalah feature yang dimiliki yang dimiliki oleh S7-300 antara lain adalah sebagai berikut:

- Memory, memiliki main memory 1400KB (CPU319), Micro Memory Card 8MB, dan backup data dan program dapat dilakukan melalui MMC.
- Dapat diekspand sampai 32 modul
- Memiliki area address I/O hingga 8192 bytes
- Language programming, KOP(LD), FUP (FBD), AWL (IL), S7-Graph (SFC), S7-SCL (ST) S7-HiGraph, dan CFC

- Communication
 - a. MPI
 - b. PtP via CP
 - c. AS-Interface via CP
 - d. Profibus via CP
 - e. Profinet via CP
 - f. Industrial Ethernet
- Interface Module (IM), IM 360/IM361 dan IM365 memungkinkan untuk membuat konfigurasi bertingkat. Bus dari interface module akan membentuk loop antara tingkat yang satu dengan tingkat berikutnya.
- Dummy Module (DM), menyediakan slot untuk signal module yang mana parameternya belum pernah ditetapkan, atau bias juga digunakan untuk menyediakan slot untuk instalasi interface module.
- Function Module (FM), memainkan fungsi special seperti counting, positioning, dan close-loop control.
- Communication Processor (CP), menyediakan fasilitas jaringan seperti PPI connection, profibus, dan industrial Ethernet.

1. Struktur Pemrograman PLC

Programmable logic controller menyediakan berbagai macam type block yang mana user program berhubungan dengan data yang dapat disimpan. Tergantung pada keperluan proses, program dapat disusun didalam block-block yang berbeda. Berikut ini adalah type dari program block:

- a. Organization Block (OB) adalah bentuk interface diantara operating system dan user program. Seluruh program dapat disimpan didalam OB1 yang secara siklis dikenal oleh operating system (program linear) atau program dapat dipisah dan disimpan dalam beberapa block (program terstruktur). Itulah sebabnya organization block dikenal sebagai operating system.
- b. Function (FC) berisi actual user program yang fungsional. Itulah yang memungkinkan untuk membuat program fungsi yang kompleks sehingga

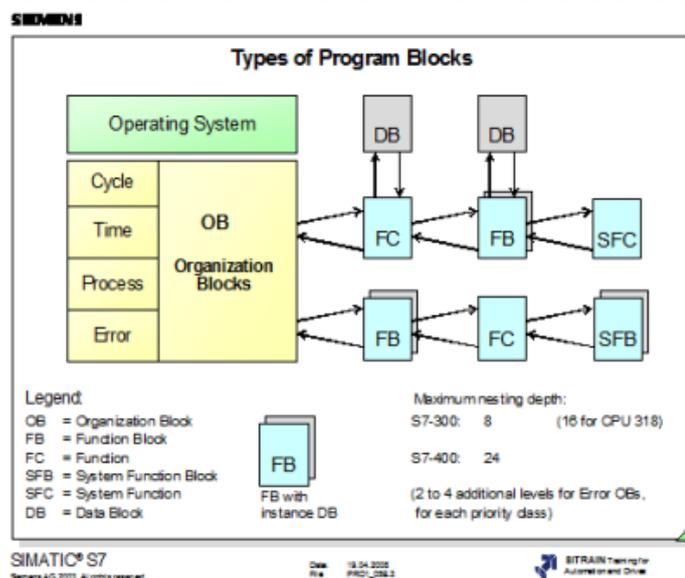
dengan begitu dapat ditugaskan oleh parameter. Sebagai hasilnya, fungsi juga cocok untuk pemrograman terulang, pemrograman fungsional yang kompleks seperti kalkulasi.

c. Sistem function (SFC) adalah integrasi fungsi parameter-assignable didalam operating system CPU.

d. Function block (FB). Secara mendasar function block (FB) menawarkan kemampuan yang sama seperti fungsi. Sebagai tambahan, function block mempunyai area memory tersendiri dalam bentuk data block. Function block cocok untuk pemrograman terulang dan pemrograman fungsional yang kompleks seperti closed-loop control.

e. System function block (SFB) adalah integrasi fungsi parameter-assignable didalam operating system CPU. Fungsi dan kemampuannya telah ditetapkan.

f. Data block (DB) adalah area data dari user program dimana data user diatur secara terstruktur.



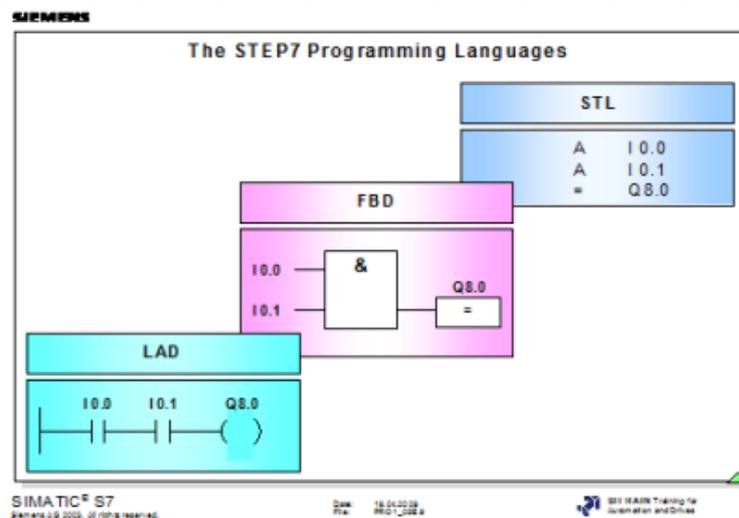
Gambar 10 tipe blok diagram

Ada beberapa pilihan bahasa pemrograman yang dapat digunakan didalam STEP 7 yaitu:

1. Ladder Diagram (LAD), yaitu bahasa pemrograman yang mirip dengan diagram rangkaian. Bahasa pemrograman ini sering menjadi daya tarik bagi

pemrogram yang mempunyai background sebagai drafting dan electrical, karena menggunakan symbol-symbol seperti coil, contact,dll.

2. Statment List (STL), yang terdiri dari kumpulan statment instruksi STEP 7. Bahasa pemrograman ini lebih disukai oleh proramer yang familiar menggunakan berbagai bahasa pemrograman.
3. Function Block Diagram (FBD), yaitu bahasa pemrograman yang menggunakan box-box fungsi. FBD memberi keuntungan dapat digunakan oleh “non-programmer” karena setiap box-box telah mengindikasikan fungsi tertentu seperti operasi fungsi logika.



Gambar 11 bahasa pemrograman

Berikut ini akan dibahas struktur pemrograman PLC menggunakan ladder diagram (LAD).

- **Bit Logic Instructions**

Instruksi bit logic bekerja dalam dua digit (0 dan 1), yang merupakan bentuk sistem bilangan biner. Bit 1 mengindikasikan keadaan aktif (energize) dan bit 0 mengindikasikan keadaan tidak aktif (deenergize). Instruksi bit logic menginterpretasikan keadaan signal 1 dan 0 dan kombinasinya menurut aljabar boolean. Hasil kombinasinya juga menghasilkan 1 dan 0 yang disebut RLO (result of logic operation). Operasi logic yang ditrigger oleh bit logic memainkan berbagai fungsi. Ada beberapa intruksi bit logic yang memainkan fungsi-fungsi berikut ini:

➤ —| |—Normally Open Contact (Address)

Symbol

<address>

—| |—

Parameter	Data Type	Memory Area	Description
<address>	BOOL	I, Q, M, L, D, T, C	Checked bit

Normally Open Contact ditutup ketika nilai bit yang tersimpan pada alamat yang telah ditetapkan oleh <address> sama dengan “1”. Ketika contact ditutup, aliran daya mengalir tepat melewati contact dan RLO bernilai “1”. Sebaliknya, ketika keadaan signal yang dispesifikasikan oleh <address> adalah “0”, contact terbuka. Ketika contact terbuka, daya tidak mengalir melewati contact dan RLO bernilai “0”. Ketika dirangkai seri, NOC dihubungkan ke bit RLO oleh logika AND. Sebaliknya bila dirangkai parallel, NOC dihubungkan ke bit RLO oleh logika OR.

Status word

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
writes:	-	-	-	-	-	X	X	X	1

➤ —| / |—Normally Closed Contact (address)

Symbol

<address>

—| / |—

Parameter	Data Type	Memory Area	Description
<address>	BOOL	I, Q, M, L, D, T, C	Checked bit

Normally Closed Contact ditutup ketika nilai bit yang tersimpan pada alamat yang telah ditetapkan oleh <address> sama dengan “0”. Ketika contact ditutup, aliran daya mengalir tepat melewati contact dan RLO bernilai “1”. Sebaliknya, ketika keadaan signal yang dispesifikasikan oleh <address> adalah “1”, contact terbuka. Ketika contact terbuka, daya tidak mengalir melewati contact dan RLO bernilai “0”. Ketika dirangkai seri, NCC dihubungkan ke bit RLO oleh logika

AND. Sebaliknya bila dirangkai parallel, NCC dihubungkan ke bit RLO oleh logika OR.

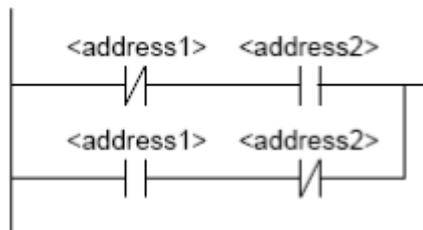
Status word

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
writes:	-	-	-	-	-	X	X	X	1

➤ **XOR function (Exclusive OR)**

XOR function merupakan sebuah jaringan NOC dan NCC yang harus dirangkai seperti ditunjukkan pada gambar berikut:

Symbols



XOR (bit exclusive OR) membuat RLO bernilai “1” jika keadaan signal jika 2 bit

yang ditetapkan mempunyai nilai yang berbeda.

Parameter	Data Type	Memory Area	Description
<address1>	BOOL	I, Q, M, L, D, T, C	Scanned bit
<address2>	BOOL	I, Q, M, L, D, T, C	Scanned bit

➤ **—|NOT|—Invert Power flow**

Symbol



Invert power flow yang menegasi bit RLO. Bila keadaan signal input adalah “1” maka keadaan signal output adalah “0”, begitu juga sebaliknya jika keadaan signal input adalah “0” maka keadaan signal output adalah “1”.

➤ **—() Output Coil**

Symbol

<address>

--()

Parameter	Data Type	Memory Area	Description
<address>	BOOL	I, Q, M, L, D	Assigned bit

Output coil bekerja seperti coil dalam diagram logic relay. Jika ada aliran daya menuju coil (RLO = 1), bit lokasi <address> diset "1". Jika tidak ada aliran daya ke coil (RLO = 0), bit lokasi <address> diset "0". Output coil hanya dapat diletakkan pada ladder sebelah kanan paling akhir. Multiple element output dimungkinkan (maksimum 16 element output). MCR (Master Control Relay) dependency diaktifkan hanya jika output coil ditempatkan disebelah zona MCR aktif. Sampai aktifnya zona MCR, MCR "ON" dan ada aliran daya ke output coil, dan bit alamat diset status aktif dari power flow. Jika MCR "off", logic "0" ditulis ke alamat yang telah dispesifikasikan tidak peduli apapun status aliran daya.

Status word

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
writes:	-	-	-	-	-	0	X	-	0

➤ **—(#)—Midline Output**

Symbol

<address>

--(#)--

Parameter	Data Type	Memory Area	Description
<address>	BOOL	I, Q, M, *L, D	Assigned bit

Midline output adalah sebuah penugasan intermediate element yang mana menyimpan bit RLO (status power flow) ke alamat yang ditetapkan <address>. Element midline output menyimpan hasil operasi logika dari element percabangan terdahulu. Bila dirangkai seri dengan contact, midline output dimasukkan seperti contact. Element midline output mungkin tidak pernah dihubungkan dengan ke power rail atau langsung setelah koneksi percabangan atau pada akhir setelah

percabangan. Negasi midline output dapat dibuat menggunakan invert power flow.

Status word

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
writes:	-	-	-	-	-	0	X	-	1

➤ **—(R) Reset Coil**

Symbol

<address>

--(R)

Parameter	Data Type	Memory Area	Description
<address>	BOOL	I, Q, M, L, D, T, C	Reset bit

Reset coil hanya dieksekusi jika RLO (aliran daya ke coil) dari instruksi terdahulu bernilai “1”. Jika aliran daya ke coil (RLO = “1”), alamat yang telah ditetapkan <address> dari element direset ke “0”. Jika tidak ada aliran daya ke coil (RLO =”0”) tidak mempunyai effect dan keadaan element tidak diubah. <address> mungkin juga menjadi timer/counter yang mana nilai timer/counternya direset “0”.

Status word

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
writes:	-	-	-	-	-	0	X	-	0

➤ **—(S) Set Coil**

Symbol

<address>

--(S)

Parameter	Data Type	Memory Area	Description
<address>	BOOL	I, Q, M, L, D	Set bit

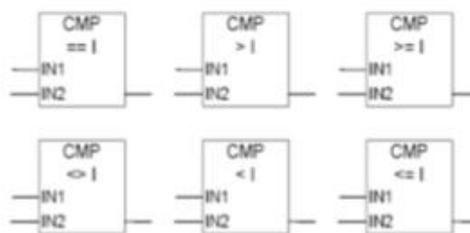
Set coil hanya dieksekusi jika RLO (aliran daya ke coil) dari instruksi terdahulu bernilai “1”. Jika RLO bernilai “1”, <address> dari element diset

“1”. RLO tidak mempunyai effect apapun jika bernilai “0” dan keadaan <address> dari element tidak diubah.

- **Compare Instructions**

- **CMP?I Compare Integer**

Symbols



Parameter	Data Type	Memory Area	Description
box input	BOOL	I, Q, M, L, D	Result of the previous logic operation
box output	BOOL	I, Q, M, L, D	Result of the comparison, is only processed further if the RLO at the box input = 1
IN1	INT	I, Q, M, L, D or constant	First value to compare
IN2	INT	I, Q, M, L, D or constant	Second value to compare

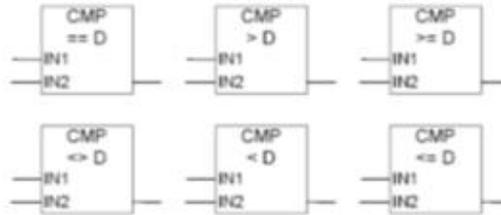
Compare integer dapat digunakan seperti sebuah contact biasa. Dia dapat diletakkan di posisi mana saja dimana contact dapat diletakkan. IN1 dan IN2 dibandingkan menurut type perbandingan (==,<>,<,>,<=,>=) yang dipilih. Jika perbandingan benar, RLO dari fungsi bernilai “1”. Jika box dirangkai seri, maka box dihubungkan ke RLO oleh operasi logika AND, sedangkan jika box dirangkai paralel maka box dihubungkan ke RLO oleh operasi logika OR.

Status word

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
writes:	x	x	x	0	-	0	x	x	1

➤ CMP?D Compare Double Integer

Symbols



Parameter	Data Type	Memory Area	Description
box input	BOOL	I, Q, M, L, D	Result of the previous logic operation
box output	BOOL	I, Q, M, L, D	Result of the comparison, is only processed further if the RLO at the box input = 1
IN1	DINT	I, Q, M, L, D or constant	First value to compare
IN2	DINT	I, Q, M, L, D or constant	Second value to compare

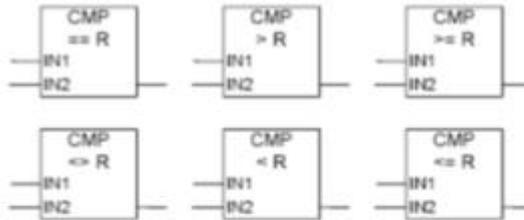
Compare double integer dapat digunakan seperti sebuah contact biasa. Dia dapat dilokasikan di posisi mana saja dimana contact dapat diletakkan. IN1 dan IN2 dibandingkan menurut type perbandingan (==,<>,<,>,<=,>=) yang dipilih. Jika perbandingan benar, RLO dari fungsi bernilai "1". Jika box dirangkai seri, maka box dihubungkan ke RLO oleh operasi logika AND, sedangkan jika box dirangkai paralel maka box dihubungkan ke RLO oleh operasi logika OR.

Status word

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
writes:	x	x	x	0	-	0	x	x	1

➤ CMP?R Compare Real

Symbols



Parameter	Data Type	Memory Area	Description
box input	BOOL	I, Q, M, L, D	Result of the previous logic operation
box output	BOOL	I, Q, M, L, D	Result of the comparison, is only processed further if the RLO at the box input = 1
IN1	REAL	I, Q, M, L, D or constant	First value to compare
IN2	REAL	I, Q, M, L, D or constant	Second value to compare

Compare real dapat digunakan seperti sebuah contact biasa. Dia dapat dilokasikan di posisi mana saja dimana contact dapat diletakkan. IN1 dan IN2 dibandingkan menurut type perbandingan (==,<>,<,>,<=,>=) yang dipilih. Jika perbandingan benar, RLO dari fungsi bernilai “1”. Jika box dirangkai seri, maka box dihubungkan ke RLO oleh operasi logika AND, sedangkan jika box dirangkai paralel maka box dihubungkan ke RLO oleh operasi logika OR.

Status word

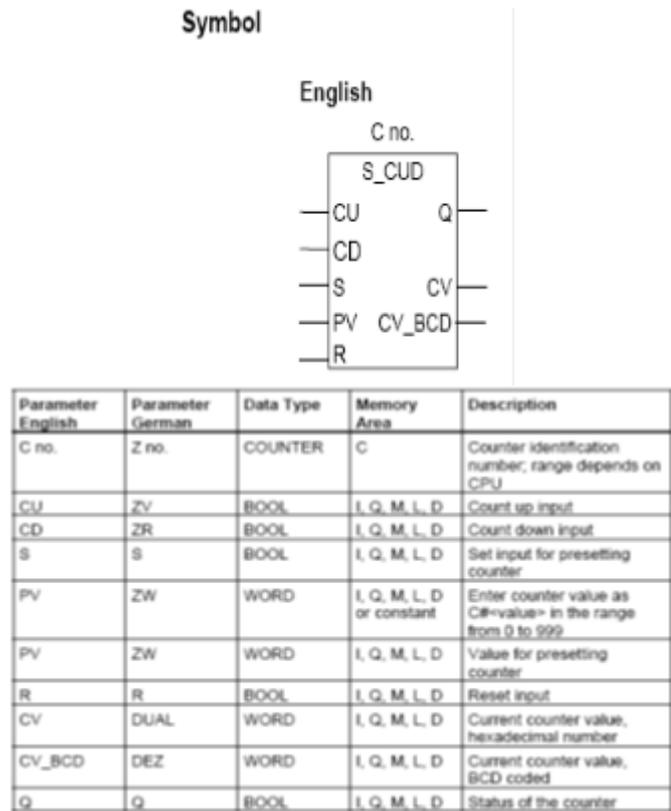
	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
writes:	x	x	x	x	x	0	x	x	1

● **Counter Instruction**

Counter mempunyai area cadangan di dalam memory CPU. Untuk tiap address counter mempunyai area memory satu word (16 bit). Set instruksi logika ladder mendukung sampai 256 counter. Instruksi counter hanyalah fungsi yang mempunyai akses ke area memory counter. Bit 0 sampai 9 dari word counter berisi nilai perhitungan dalam kode biner.. Nilai perhitungan dipindahkan ke word

counter ketika counter diset. Counter memiliki nilai range perhitungan mulai dari 0 sampai 999. Berikut ini adalah berbagai instruksi counter:

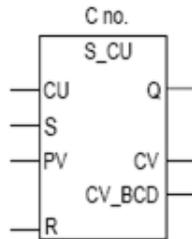
➤ S_CUD Up-Down Counter



S_CUD (up-down counter) ditetapkan lebih dahulu nilainya oleh input PV jika ada positive edge pada input S (keadaan input signal S berubah dari “0” ke “1”). Jika input R bernilai “1”, counter direset dan perhitungan diset ke nol. Counter dinaikan satu nilainya (CV bertambah 1) jika keadaan signal pada input CU berubah dari “0” ke “1” sampai nilainya kurang dari 999. Counter nilainya diturunkan 1 (CV berkurang 1) jika ada positive edge pada input CD (keadaan input signal CD berubah dari “0” ke “1”) sampai nilainya masih lebih besar dari nol. Jika kedua input count keduanya mengalami positive edge, maka kedua intruksi (up and down counter) akan dieksekusi sehingga nilai perhitungan tidak berubah (CV tetap). Keadaan signal output Q akan bernilai “1” jika perhitungan lebih besar dari nol dan akan bernilai “0” jika perhitungan sama dengan nol.

➤ S_CU Up Counter

Symbol

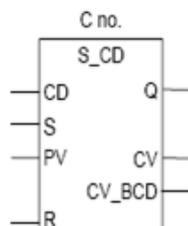


Parameter English	Parameter German	Data Type	Memory Area	Description
C no.	Z no.	COUNTER	C	Counter identification number, range depends of CPU
CU	ZV	BOOL	I, Q, M, L, D	Count up input
S	S	BOOL	I, Q, M, L, D	Set input for presetting counter
PV	ZW	WORD	I, Q, M, L, D or constant	Enter counter value as C#-value> in the range from 0 to 999
PV	ZW	WORD	I, Q, M, L, D	Value for presetting counter
R	R	BOOL	I, Q, M, L, D	Reset input
CV	DUAL	WORD	I, Q, M, L, D	Current counter value, hexadecimal number
CV_BCD	DEZ	WORD	I, Q, M, L, D	Current counter value, BCD coded
Q	Q	BOOL	I, Q, M, L, D	Status of the counter

S_CU (up counter) ditetapkan lebih dahulu nilainya oleh PV jika ada positive edge (keadaan signal berubah dari “0” ke “1”) pada input S. Counter direset jika input R bernilai “1” dan CV diset ke nol. Counter dinaikan satu nilainya (CV bertambah 1) jika keadaan signal input CU berubah dari “0” ke “1” sampai nilainya kurang dari 999. Keadaan signal output Q akan bernilai “1” jika perhitungan lebih besar dari nol dan akan bernilai “0” jika perhitungan sama dengan nol.

➤ S_CD Down Counter

Symbol



Parameter English	Parameter German	Data Type	Memory Area	Description
C no.	Z no.	COUNTER	C	Counter identification number, range depends of CPU
CD	ZR	BOOL	I, Q, M, L, D	Count down input
S	S	BOOL	I, Q, M, L, D	Set input for presetting counter
PV	ZW	WORD	I, Q, M, L, D or constant	Enter counter value as C#-value> in the range from 0 to 999
PV	ZW	WORD	I, Q, M, L, D	Value for presetting counter
R	R	BOOL	I, Q, M, L, D	Reset input
CV	DUAL	WORD	I, Q, M, L, D	Current counter value, hexadecimal number
CV_BCD	DEZ	WORD	I, Q, M, L, D	Current counter value, BCD coded
Q	Q	BOOL	I, Q, M, L, D	Status counter

S_CD (down counter) ditetapkan lebih dahulu nilainya oleh PV jika ada positive edge (keadaan signal berubah dari “0” ke “1”) pada input S. Counter direset jika input R bernilai “1” dan CV diset ke nol. Counter diturunkan satu nilainya (CV berkurang 1) jika keadaan signal input CU berubah dari “0” ke “1”

sampai nilainya masih lebih besar dari 0. Keadaan signal output Q akan bernilai “1” jika perhitungan lebih besar dari nol dan akan bernilai “0” jika perhitungan sama dengan nol.

➤ —(SC) Set Counter Value

Symbol

<C no.>
 ---(SC)
 <preset value>

Parameter English	Parameter German	Data Type	Memory Area	Description
<C no.>	<Z no.>	COUNTER	C	Counter number to be preset
<preset value>	<preset value>	WORD	I, Q, M, L, D or constant	Value for presetting BCD (0 to 999)

Set Counter Value mengeksekusi hanya jika ada positive edge pada RLO (RLO berubah nilainya dari “0” ke “1”). Pada saat itu, preset value (PV) mentransfer kedalam specified counter.

➤ —(CU) Up Counter Coil

Symbol

<C no.>
 ---(CU)

Parameter English	Parameter German	Data Type	Memory Area	Description
<C no.>	<Z no.>	COUNTER	C	Counter identification number, range depends on CPU

Up Counter Coil menaikkan satu nilai dari specified counter jika ada positive edge pada RLO sampai nilai counter tidak lebih dari 999. Jikadidak ada positif edge pada RLO atau nilai counter sudah mencapai nilai 999 maka nilai counter tidak akan berubah.

➤ —(CD) Down Counter Coil

Symbol

	Parameter English	Parameter German	Data Type	Memory Area	Description
<C no.>	<C no.>	<Z no.>	COUNTER	C	Counter identification number; range depends on CPU
---(CD)					

Down Counter Coil menurunkan satu nilai dari specified counter jika ada positive edge pada RLO sampai nilai counter masih lebih besar dari nol. Jikatidak ada positif edge pada RLO atau nilai counter bernilai nol maka nilai counter tidak akan berubah.

● **Logic Control Instructions**

Logic control logic dapat digunakan disemua block logika, baik organization block (OB), function block (FB), maupun function (FC). Label adalah pengalamatan dari instruksi jump. Label maksimum terdiri dari empat karakter, karakter pertama harus huruf sedangkan karakter yang lainnya bisa huruf/angka. Label jump mengindikasikan tujuan kemana program akan melompat. Berikut ini adalah beberapa logic control instruction:

➤ —(JMP) Unconditional Jump

Symbol

<label name>
---(JMP)

—(JMP) adalah fungsi seperti absolute jump ketika tidak ada element ladder lain antara sebelah kiri power rail dan instruksi. Tujuan instruksi jump (label) juga harus exist untuk setiap intruksi —(JMP). Semua instruksi diantara instruksi jump dan label tidak akan dieksekusi.

➤ —(JMP) Conditional Jump

Symbol

<label name>

---(JMP)

—(JMP) adalah fungsi sebagai conditional jump ketika RLO dari operasi logika sebelumnya bernilai “1”. Tujuan instruksi jump (label) juga harus exist untuk setiap intruksi —(JMP). Semua instruksi diantara instruksi jump dan label tidak akan dieksekusi. Ketika conditional jump tidak dieksekusi, RLO berubah menjadi “1” setelah instruksi jump.

➤ —(JMPN) Jump if Not

Symbol

<label name>

---(JMPN)

—(JMPN) adalah fungsi jump yang mana akan dijalankan ketika RLO bernilai “0”. Label juga harus exist untuk setiap intruksi —(JMPN). Semua instruksi diantara instruksi jump dan label tidak akan dieksekusi. Ketika conditional jump tidak dieksekusi, RLO berubah menjadi “1” setelah instruksi jump.

3D	Pivot to 2nd Idler	600 mm
3I	Pivot to Test Weight Height	0
3L	Pivot to Test Weight Length	1200
3M	Pivot to Weighbridge Height	160
3N	Roll to Weighbridge Height	250 mm (Actual 290 mm)
3O	Number of Load Cells	2
4	Idler Spacing	1200 mm
5	Conveyor's Angle Degrees	12.815 Deg
6	Load Cell Capacity	226.8 Kgs
7	Load Cell Sensitivity	3.000 mv/v
8A	Load Cell # 1 Resistance	350 Ohm
8B	Load Cell # 2 Resistance	350 Ohm
9	Speed Input	Single
III	CALIBRATION DATA SCROLL	
1	Calibration Mode	Weight
6	Total Test Weights	200 KGS
7	Weight- Cal. Constant	126.09