



BAB II

TINJAUAN PUSTAKA

2.1 Programmable Logic Controllers (PLC)

Programmable Logic Controllers (PLC) adalah komputer elektronik yang mudah digunakan (*user friendly*) yang memiliki fungsi kendali untuk berbagai tipe dan tingkat kesulitan yang beraneka ragam. 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 logic (ALU), yakni melakukan operasi membandingkan, menjumlahkan, mengalikan, membagi, mengurangi, negasi, AND, OR, dan lain sebagainya.
- c. *Controller*, menunjukkan kemampuan dalam mengontrol dan mengatur proses sehingga menghasilkan output yang diinginkan.^[1]

Menurut National Electrical Manufacturing Assosiation (NEMA) PLC didefinisikan sebagai suatu perangkat elektronik digital dengan memori yang dapat diprogram untuk menyimpan instruksi-instruksi yang menjalankan fungsi-fungsi spesifik seperti: logika, sekuen, *timing*, *counting*, dan aritmatika untuk mengontrol suatu mesin industri atau proses industri sesuai dengan yang diinginkan. PLC mampu mengerjakan suatu proses terus menerus sesuai variabel masukan dan memberikan keputusan sesuai keinginan pemrograman sehingga nilai keluaran tetap terkontrol.^[2]

Pengendali logika Terprogram pada dasarnya adalah sebuah komputer yang khusus di rancang untuk mengontrol suatu proses atau mesin. Proses yang dikontrol ini dapat berupa regulasi Variabel secara kontinu seperti pada sistem-sistem servo, atau hanya melibatkan dua kontrol keadaan (On/Off) Saja, tetapi dilakukan secara berulang ulang seperti umum di jumpai pada proses pengepakan, sistem konveyor, penanganan bahan, perakitan otomasi dan lain sebagainya.



Walaupun istilah PLC secara bahasa berarti pengontrol logika yang dapat di program, tetapi pada kegunaanya, PLC secara fungsional tidak lagi terbatas pada fungsi-fungsi logika saja. Sebuah PLC dewasa ini dapat melakukan perhitungan-perhitungan aritmatika yang relatif kompleks, fungsi komunikasi, dokumentasi, dan lain sebagainya.

Berdasarkan jumlah input/output PLC dapat di bagi menjadi tiga kelompok besar :

1. PLC Mikro. PLC dapat dikategorikan mikro jika jumlah input/output pada PLC kurang dari 32 terminal.
2. PLC Mini. Kategori ukuran mini ini adalah jika PLC tersebut memiliki jumlah input/output antara 32 sampai 128 terminal.
3. PLC Large. PLC ukuran ini dikenal juga dengan PLC tipe Rack. PLC dapat di kategorikan PLC rack jika input/output-nya lebih dari 128 terminal.^[3]

Dengan demikian, semakin kompleks proses yang harus ditangani maka semakin penting PLC untuk mempermudah proses-proses tersebut (dan sekaligus menggantikan beberapa alat yang diperlukan). Selain itu sistem kontrol proses konvensional memiliki beberapa kelemahan antara lain :

1. Perlu kerja keras saat dilakukan pengkabelan.
2. Kesulitan saat dilakukan penggantian dan perubahan.
3. Kesulitan saat dilakukan pelacakan kesalahan.
4. Saat terjadi masalah, waktu tunggu menentu dan biasanya lama.

Sedangkan dengan penggunaan PLC memiliki beberapa kelebihan dibandingkan dengan sistem kontrol konvensional antara lain :

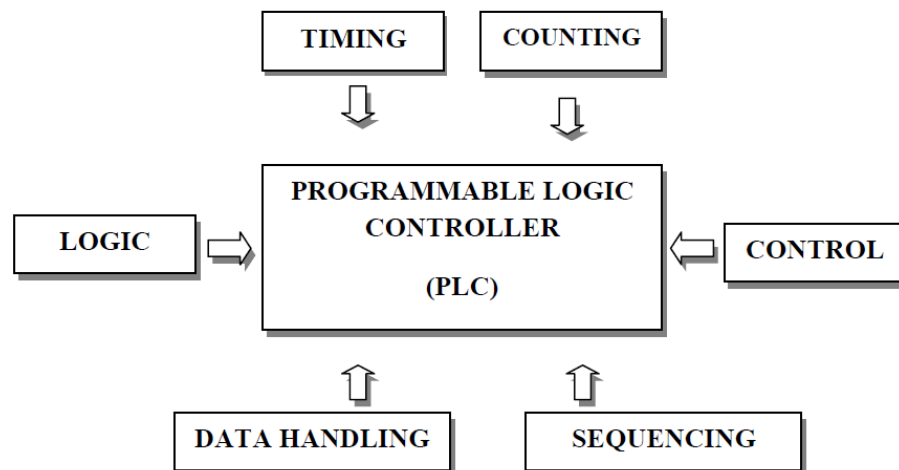
1. Dibandingkan dengan sistem kontrol proses konvensional, jumlah kabel yang dibutuhkan bisa berkurang hingga 80%.
2. PLC mengkonsumsi daya lebih rendah dibandingkan dengan sistem kontrol proses konvensional berbasis relay.



3. Fungsi diagnostik pada sebuah kontroller PLC memperbolehkan pendekteksian kesalahan yang mudah dan cepat.
4. Perubahan pada urutan proses atau operasional dapat dilakukan dengan mudah, hanya dengan melakukan perubahan atau penggantian program, baik melalui terminal konsol maupun komputer PC.
5. Tidak membutuhkan spare part yang banyak.
6. Lebih murah dibandingkan dengan sistem konvensional, khususnya dalam kasus penggunaan instrumen I/O yang cukup banyak dan fungsi operasional prosesnya cukup kompleks.
7. Ketahanan PLC jauh lebih baik dibandingkan dengan relay otomatis mekanik.^[4]

2.2 Fungsi PLC (Programmable Logic Control)^[5]

PLC merupakan perangkat elektronik yang bekerja secara digital, yang dapat menggantikan fungsi dari relay, timer, counter dan sekaligus dapat melaksanakan fungsi aritmatik untuk mengontrol berbagai jenis mesin atau proses analog atau digital input dan output modul serta transaksi data. PLC juga memiliki memori yang dapat diprogram.



Gambar 2.1 Fungsi PLC.



2.3 Pendekatan Sistematis Dalam Perancangan Sistem Kendali Proses^[6]

Pertama, anda perlu memilih suatu instrumen atau sistem yang hendak dikontrol. Sistem yang terotomasi bisa berupa sebuah mesin atau suatu proses yang kemudian disebut sebagai sistem kendali proses. Fungsi dari sistem kendali proses ini secara terus menerus akan mengamati sinyal-sinyal yang berhasil dari piranti-piranti masukan sensor dan tanggapannya berupa suatu sinyal yang diberikan ke piranti keluaran eksternal yang secara langsung mengontrol bagaimana suatu sistem beroperasi atau bekerja.

Kedua, Anda perlu menentukan semua instrumen masukan (input) dan keluaran yang akan dihubungkan ke PLC. Piranti masukan dapat berupa saklar, sensor dan lain sebagainya. Sedangkan piranti keluaran (output) dapat berupa solenoida, kran elektromagnet, motor, relai, starter magnet begitu juga dengan instrumen lain yang bisa menghasilkan suara atau cahaya lampu dan lain sebagainya.

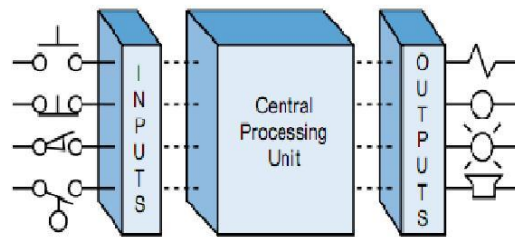
Ketiga, membuat program yang lebih dikenal dengan diagram tangga (untuk PLC) sesuai dengan jalannya proses yang diinginkan. Dalam hal ini bisa digunakan terminal konsol yang dapat berhubungan langsung dengan PLC atau melalui komputer PC yang memiliki saluran komunikasi yang dibutuhkan untuk mentransfer program dari komputer PC ke PLC maupun sebaliknya.

Keempat, program disimpan ke dalam PLC, baik dilakukan secara langsung melalui terminal konsol maupun melalui komputer PC.

2.4 Prinsip Kerja PLC^[6]

Secara umum, PLC terdiri dari dua komponen penyusun utama (Gambar 2.2) :

- Central Processing Unit (CPU)
- Sistem antarmuka input/output

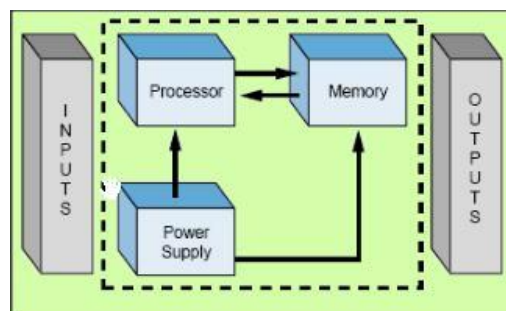


Gambar 2.2 Blok diagram PLC.

Fungsi dari CPU adalah mengatur semua proses yang terjadi di PLC. Ada tiga komponen utama penyusun CPU ini.

- Prosesor
- Memori
- Power supply

Interaksi antara ketiga komponen ini dapat di lihat pada gambar 2.3



Gambar 2.3 Blok diagram CPU pada PLC.

Pada dasarnya, operasi PLC ini relatif sederhana : peralatan luar dikoneksikan dengan modul input/output PLC yang tersedia. Peralatan ini dapat berupa sensor-sensor analog, push button, limit switch, proximity, encoder, motor starter, solenoid, lampu, dan lain sebagainya.

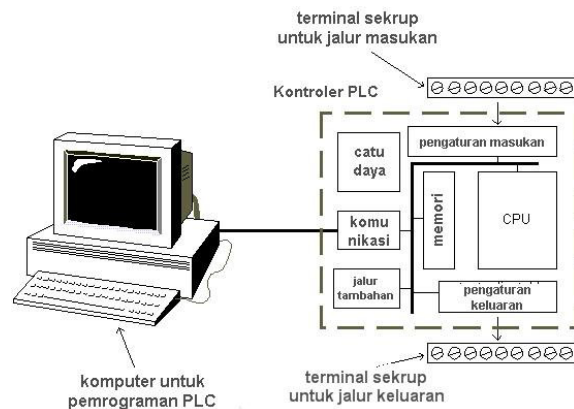
Berkaitan dengan pemrograman PLC ini, sebenarnya ada lima model atau metode yang telah distandarisasi penggunaannya oleh IEC (International Electrical Commission) adalah sebagai berikut :



1. List Instruksi (Instruction List) – Pemrograman dengan menggunakan instruksi-instruksi bahasa level rendah (Mnemonic), seperti LD/STR, NOT, AND dan lain sebagainya.
2. Diagram Ladder (Ladder Diagram) – Pemrograman berbasis logika relai, cocok digunakan untuk persoalan-persoalan kontrol diskret yang input/output hanya memiliki dua kondisi On atau Off seperti pada sistem kontrol konveyor, lift, dan motor-motor industri.
3. Diagram Blok Fungsional (Function Blok Diagram) – pemrograman berbasis aliran data secara grafis. Banyak digunakan untuk tujuan kontrol proses yang melibatkan perhitungan-perhitungan kompleks dan akuisisi data analog.
4. Diagram Fungsi Sekuensial (Sequential Function Charts) – metode grafis untuk pemrograman terstruktur yang banyak melibatkan langkah-langkah rumit, seperti pada bidang robotika, perakitan kendaraan, Batch Control, dan lain sebagainya.
5. Teks Tertekstur (Structured Text) – Tidak seperti keempat metode sebelumnya, pemrograman ini menggunakan statement-statement yang umum dijumpai pada bahasa level tinggi (high level programming) seperti If/Then, Do/while, Case, For/Next, dan lain sebagainya. Dalam aplikasinya, model ini cocok digunakan untuk perhitungan-perhitungan matematis yang kompleks, pemrosesan tabel dan data, serta fungsi kontrol yang memerlukan algoritma khusus.

2.5 Komponen – Komponen PLC^[7]

PLC sesungguhnya merupakan sistem mikrokontroler khusus untuk industri, artinya seperangkat perangkat lunak dan keras yang diadaptasi untuk keperluan aplikasi dalam dunia industri. Elemen-elemen dasar sebuah PLC ditunjukkan pada gambar berikut.



Gambar 2.4 Bagian-bagian PLC.

a. Unit Pengolah Pusat atau Central Processing Unit (CPU)

Unit pengolah pusat atau CPU merupakan otak dari sebuah kontroler PLC. CPU itu sendiri biasanya merupakan sebuah mikrokontroler (versi mini mikrokomputer lengkap).

b. Memori

Memori sistem digunakan oleh PLC untuk sistem kontrol proses. Selain berfungsi untuk menyimpan sistem operasi juga digunakan untuk menyimpan program yang harus dijalankan, dalam bentuk biner, hasil terjemahan diagram tangga yang dibuat oleh pengguna atau pemrogram.

c. Pemrograman PLC

Kontroler PLC dapat diprogram melalui komputer dan dapat juga diprogram melalui pemrograman manual yang biasa disebut konsol (*console*). Untuk keperluan ini dibutuhkan perangkat lunak, yang biasanya juga tergantung pada produk PLC-nya.

d. Catu Daya PLC

Catu daya listrik digunakan untuk memberikan pasokan catu daya ke seluruh bagian PLC (termasuk CPU, memori, dan lain-lain). Kebanyakan PLC bekerja dengan catu daya 24 VDC atau 220 VAC. Beberapa PLC besar catu dayanya terpisah, sedangkan yang medium atau kecil catu dayanya sudah menyatu.

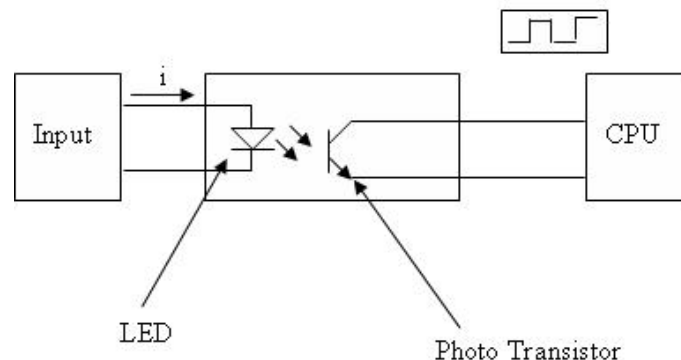


e. Masukan-masukan PLC

Kecerdasan sebuah sistem terotomasi sangat tergantung pada kemampuan sebuah PLC dalam membaca sinyal dari berbagai macam jenis sensor dan piranti-piranti masukan lainnya seperti Limit Switch dan Push Button.

f. Pengaturan atau Antarmuka Masukan

Antarmuka masukan berada diantara jalur masukan yang sesungguhnya dengan unit CPU. Tujuannya adalah melindungi CPU dari sinyal-sinyal yang tidak dikehendaki yang bisa merusak CPU itu sendiri. Modul antarmuka masukan ini berfungsi untuk mengkonversi atau mengubah sinyal-sinyal masukan dari luar ke sinyal-sinyal yang sesuai dengan tegangankerja CPU yang bersangkutan (misalnya, masukan dari sensor dengan tegangan kerja 24 VDC harus diionversikan menjadi tegangan 5 VDC agar sesuai dengan tegangan kerja CPU).



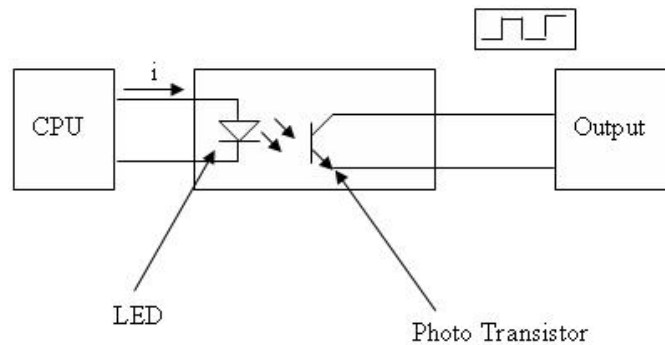
Gambar 2.5 Antarmuka Rangkaian Input PLC.

g. Keluaran-Keluaran PLC

Sistem terotomasi tidak lengkap tanpa adanya fasilitas keluaran atau fasilitas untuk menghubungkan dengan alat-alat eksternal (yang dikendalikan). Beberapa alat atau piranti yang banyak digunakan adalah motor, solenoida, relai, lampu indikator, speaker, alarm dan lain sebagainya.

h. Pengaturan atau Antarmuka Keluaran

Sebagaimana pada antarmuka masukan, keluaran juga membutuhkan antarmuka yang sama yang digunakan untuk memberikan perlindungan antara CPU dengan peralatan eksternal.



Gambar 2.6 Antarmuka Rangkaian Output PLC.

i. Jalur Ekstensi atau Tambahan

PLC umumnya memiliki jumlah masukan dan keluaran yang terbatas. Jika diinginkan, jumlah ini dapat ditambahkan menggunakan sebuah modul keluaran dan masukan tambahan (*I/O expansion atau I/O extension module*).

2.6 Struktur Memori PLC OMRON^[7]

Pengenalan terhadap struktur memori suatu PLC sangat diperlukan, agar supaya dapat menggunakannya dengan tepat sesuai dengan fungsinya masing-masing. Pada PLC Omron CPM1 terdapat beberapa memori yang memiliki fungsi-fungsi khusus. Masing-masing lokasi memori memiliki ukuran 16-bit atau 1 word, beberapa word membentuk daerah atau region dan masing-masing region inilah yang memiliki fungsi-fungsi khusus. Tidak seperti mikrokontroler, yang hanya mendefinisikan sebagian dari fungsi-fungsi memorinya, pada PLC semua bagian memori didefinisikan fungsinya secara khusus. Selain itu dalam PLC semua lokasi memori dapat dialamati per-bit, atau dengan kata lain dapat diakses per-bit atau *bit addressable*. Tabel 1 memperlihatkan Data Area PLC Omron CPM1.

Daerah IR

Bagian memori ini digunakan untuk menyimpan status keluaran dan masukan PLC. Beberapa bit berhubungan langsung dengan terminal masukan (input) dan keluaran (output). Daerah memori IR terbagi atas tiga macam area, yaitu area masukan (*input area*), area keluaran (*output area*), dan area kerja (*work area*). Untuk mengakses memori ini, cukup dengan menuliskan angkanya saja,



000 untuk masukan, 010 untuk keluaran dan 200 untuk memori kerja. Tabel 1 memperlihatkan terminal unit CPU yang digunakan untuk input/output eksternal dari beberapa model PLC CPM1.

Tabel 2.1 Data Area PLC Omron CPM1.

Data area		Words	Bits	Function
IR area ¹	Input area	IR 000 to IR 009 (10 words)	IR 00000 to IR 00915 (160 bits)	These bits can be allocated to the external I/O terminals.
	Output area	IR 010 to IR 019 (10 words)	IR 01000 to IR 01915 (160 bits)	
	Work area	IR 200 to IR 231 (32 words)	IR 20000 to IR 23115 (512 bits)	Work bits can be freely used within the program.
SR area		SR 232 to SR 255 (24 words)	SR 23200 to SR 25515 (384 bits)	These bits serve specific functions such as flags and control bits.
TR area		---	TR 0 to TR 7 (8 bits)	These bits are used to temporarily store ON/OFF status at program branches.
HR area ²		HR 00 to HR 19 (20 words)	HR 0000 to HR 1915 (320 bits)	These bits store data and retain their ON/OFF status when power is turned off.
AR area ²		AR 00 to AR 15 (16 words)	AR 0000 to AR 1515 (256 bits)	These bits serve specific functions such as flags and control bits.
LR area ¹		LR 00 to LR 15 (16 words)	LR 0000 to LR 1515 (256 bits)	Used for a 1:1 data link with another PC.
Timer/Counter area ²		TC 000 to TC 127 (timer/counter numbers) ³		The same numbers are used for both timers and counters.
DM area	Read/write ²	DM 0000 to DM 0999 DM 1022 to DM 1023 (1,002 words)	---	DM area data can be accessed in word units only. Word values are retained when the power is turned off.
	Error log	DM 1000 to DM 1021 (22 words)	---	Used to store the time of occurrence and error code of errors that occur. These words can be used as ordinary read/write DM when the error log function isn't being used.
	Read-only ⁴	DM 6144 to DM 6599 (456 words)	---	Cannot be overwritten from program.
	PC Setup ⁴	DM 6600 to DM 6655 (56 words)	---	Used to store various parameters that control PC operation.

Tabel 2.2 IR Area PLC CPM1 yang dihubungkan dengan terminal unit CPU.

Number of I/O Terminals on the CPU Unit	CPU Unit Terminals		Expansion I/O Unit Terminals		Power Supply	Model Number
	Inputs	Outputs	Inputs	Outputs		
10	6 points: 00000 to 00005	4 points: 01000 to 01003	12 points: 00100 to 00111	8 points: 01100 to 01107	AC	CPM1-10CDR-A
					DC	CPM1-10CDR-D
20	12 points: 00000 to 00011	8 points: 01000 to 01007	12 points: 00100 to 00111	8 points: 01100 to 01107	AC	CPM1-20CDR-A
					DC	CPM1-20CDR-D
30	18 points: 00000 to 00011 00100 to 00105	12 points: 01000 to 01007 01100 to 01103	12 points: 00200 to 00211	8 points: 01200 to 01207	AC	CPM1-30CDR-A
					DC	CPM1-30CDR-D
			36 points: 00200 to 00211 00300 to 00311 00400 to 00411	24 points: 01200 to 01207 01300 to 01307 01400 to 01407	AC	CPM1-30CDR-A-V1 (Available soon)
					DC	CPM1-30CDR-D-V1 (Available soon)

Daerah SR

Daerah SR merupakan bagian khusus dari lokasi memori yang digunakan sebagai bit-bit kontrol dan status (*flag*), paling sering digunakan untuk pencacah dan interupsi. Misalnya SR253.13 (atau 253.13 saja) adalah *Always ON Flag* atau



nilainya selalu ON selama PLC dihidupkan sedangkan SR253.14 (atau 253.14 saja) adalah *Always OFF Flag* atau nilainya selalu OFF selama PLC dihidupkan. SR255.04 (atau 255.04 saja) digunakan sebagai Flag *CARRY (CY)*. SR255.05 SR255.06 dan SR255.07 masing-masing digunakan untuk menyimpan status lebih besar dari (*Greater Than*), sama dengan (*Equals*), dan lebih kecil dari (*Less Than*) hasil dari fungsi perbandingan *CMP*.

Daerah TR

Saat pindah ke sub-program selama eksekusi program maka semua data yang terkait hingga batasan *RETURN* sub-program akan disimpan dalam daerah TR ini. Untuk PLC CPM1A (juga CPM2A) hanya memiliki 8 bit yaitu TR0 sampai TR7.

Daerah HR

Bit-bit pada daerah HR ini digunakan untuk menyimpan data dan tidak akan hilang walaupun PLC sudah tidak mendapat catu daya atau PLC sudah dimatikan karena menggunakan baterai. Untuk PLC CPM1A (juga CPM2A) daerah ini terdiri dari 20 word (HR00 sampai HR19) atau 320 bit (HR00.00 sampai HR19.15). Bit-bit HR ini dapat digunakan bebas di dalam program sebagaimana bit-bit kerja (*work bits*).

Daerah AR

Ini merupakan daerah lain yang juga digunakan untuk menyimpan bit-bit kontrol dan status seperti status PLC kesalahan waktu sistem dan lain sejenisnya. Dan seperti daerah HR, daerah AR juga dilengkapi dengan baterai sehingga data-data kontrol maupun status tetap akan tersimpan walaupun PLC sudah dimatikan. Untuk PLC CPM1A daerah ini terdiri dari 16 word (AR00 sampai AR15) atau 256 bit (AR00.00 sampai AR15.15). Misalnya AR 08 bit 00 sampai 03 digunakan untuk menyimpan kode kesalahan Port RS232 dengan ketentuan tiap bit:

00 – Normal

01 – Kesalahan Paritas

02 – Kesalahan Frame

03 – Kesalahan Overrun



Daerah LR

Digunakan sebagai pertukaran data saat dilakukan koneksi atau hubungan dengan PLC yang lain. Untuk PLC CPM1A dan CPM2A daerah ini terdiri atas 16 word (LR00 sampai LR15) atau 256 bit (LR00.00 sampai LR15.15).

Daerah Pewaktu/Pencacah (*Timer/Counter*) – T/C Area

Daerah ini digunakan untuk menyimpan nilai-nilai pewaktu atau pencacah. Untuk PLC CPM1A terdapat 128 lokasi (TC000 sampai TC127) sedangkan pada PLC CPM2A terdapat 226 lokasi (TC000 sampai TC225).

Daerah DM

Berisikan data-data yang terkait dengan pengaturan komunikasi dengan komputer dan data pada saat ada kesalahan.

2.7 Instruksi-Instruksi Pada PLC Omron

2.7.1. Instruksi-instruksi tangga (*ladder instructions*)

instruksi-instruksi tangga adalah instruksi-instruksi yang terkait dengan kondisi di dalam diagram tangga. instruksi-instruksi tangga, baik yang independen maupun yang kombinasi atau gabungan dengan blok instruksi, akan membentuk kondisi eksekusi. gambar 2.7 memperlihatkan kode mnemonik, diagram tangga, dan area data

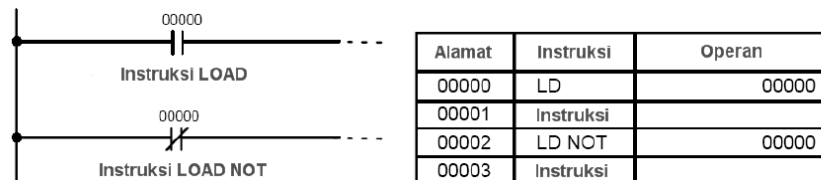
	Ladder Symbols	Operand Data Areas
LOAD – LD		B: Bit IR, SR, AR, HR, TC, LR, TR
LOAD NOT – LD NOT		B: Bit IR, SR, AR, HR, TC, LR
AND – AND		B: Bit IR, SR, AR, HR, TC, LR
AND NOT – AND NOT		B: Bit IR, SR, AR, HR, TC, LR
OR – OR		B: Bit IR, SR, AR, HR, TC, LR
OR NOT – OR NOT		B: Bit IR, SR, AR, HR, TC, LR

Gambar 2.7 Kode mneumonic, diagram tangga, dan area data operan dari instruksi-instruksi tangga.



A. LOAD (LD) dan LOAD NOT (LD NOT)

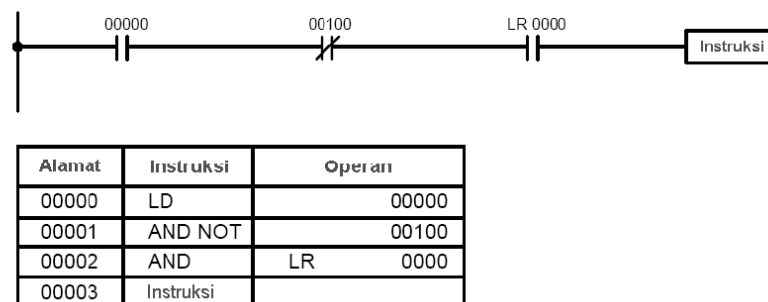
Kondisi pertama yang mengawali blok logika di dalam diagram tangga berkaitan dengan instruksi LOAD (LD) atau LOAD NOT (LD NOT). Contoh instruksi ini ditunjukkan pada gambar 2.8



Gambar 2.8 Contoh instruksi LD dan LD NOT.

B. AND dan AND NOT

Jika terdapat dua atau lebih kondisi yang dihubungkan seri pada garis instruksi yang sama maka kondisi pertama menggunakan instruksi LD atau LD NOT, dan sisanya menggunakan instruksi AND atau AND NOT. Gambar 2.9 menunjukkan suatu penggalan diagram tangga yang mengandung tiga kondisi yang dihubungkan secara seri pada garis instruksi yang sama dan berkaitan dengan instruksi LD, AND NOT, dan AND. Masing-masing instruksi tersebut membutuhkan satu baris kode mnemonic



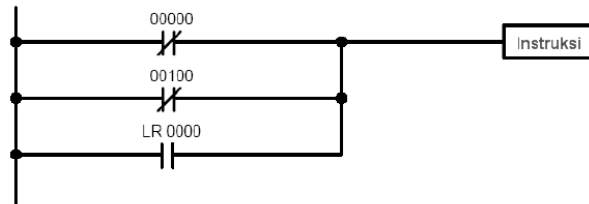
Gambar 2.9 Contoh penggunaan instruksi AND dan AND NOT.

C. OR dan OR NOT

Jika dua atau lebih kondisi yang dihubungkan paralel, artinya dalam garis instruksi yang berbeda kemudian bergabung lagi dalam satu garis instruksi yang sama maka kondisi pertama terkait dengan instruksi LD dan LD NOT dan sisanya berkaitan dengan instruksi OR dan OR NOT. Gambar 2.10 menunjukkan tiga



buah instruksi yang berkaitan dengan instruksi LD NOT, OR NOT, dan OR. Masing-masing instruksi tersebut membutuhkan satu baris kode mnemonik.

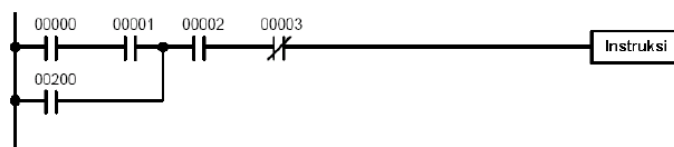


Alamat	Instruksi	Operan
00000	LD NOT	00000
00001	OR NOT	00100
00002	OR	LR 0000
00003	Instruksi	

Gambar 2.10 Contoh penggunaan instruksi OR dan OR NOT.

D. Kombinasi instruksi AND dan OR

Jika instruksi AND dan OR digabung atau dikombinasikan dalam suatu rangkaian tangga yang kompleks maka bisa dipandang satu persatu, artinya bisa dilihat masing-masing hasil gabungan dua kondisi menggunakan instruksi AND atau OR secara sendiri-sendiri kemudian menggabungkannya menjadi satu kondisi menggunakan instruksi AND atau OR yang terakhir. Gambar 2.11 menunjukkan contoh diagram tangga yang mengimplentasikan cara seperti tersebut di atas.



Alamat	Instruksi	Operan
00000	LD	00000
00001	AND	00001
00002	OR	00200
00003	AND	00002
00004	AND NOT	00003
00005	Instruksi	

Gambar 2.11 Contoh penggabungan instruksi AND dan OR.

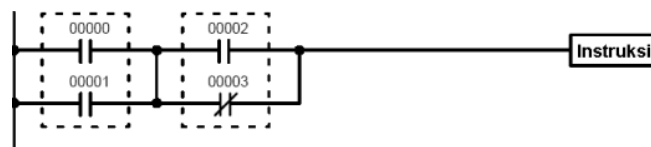


2.7.2. Instruksi-instruksi blok logika

Instruksi-instruksi blok logika tidak berhubungan dengan suatu kondisi tertentu pada diagram tangga, melainkan untuk menyatakan hubungan antar blok-blok logik, misalnya instruksi AND LD akan meng-AND-logik-kan kondisi eksekusi yang dihasilkan oleh dua blok logik, demikian juga dengan OR LD untuk meng-OR logikkan kondisi eksekusi yang dihasilkan dua blok logik.

a) AND LOAD (AND LD)

Gambar 2.12 menunjukkan contoh penggunaan blok logik AND LD yang terdiri atas dua blok logik, yang akan menghasilkan kondisi ON jika blok logik kiri dalam kondisi ON (salah satu dari IR000.00 atau IR000.01 yang ON) dan blok logik kanan juga dalam keadaan ON (IR000.02 dalam kondisi ON atau IR000.03 dalam kondisi OFF).

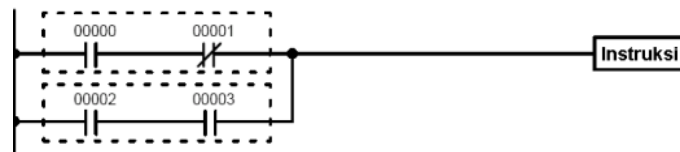


Alamat	Instruksi	Operan
00000	LD	00000
00001	OR	00001
00002	LD	00002
00003	OR NOT	00003
00004	AND LD	---

Gambar 2.12 Contoh penggunaan instruksi blok logika AND LD.

b) OR LOAD (OR LD)

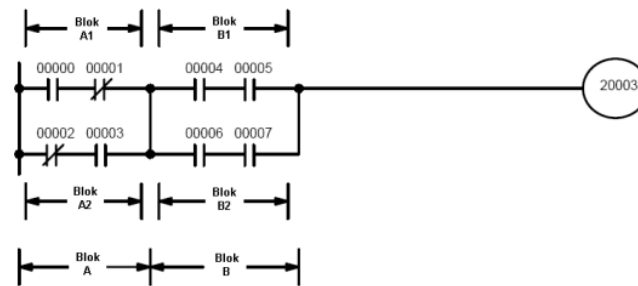
Instruksi ini digunakan untuk meng-OR-logik-kan dua blok logika. Gambar 2.13 menunjukkan contoh penggunaan blok logik OR LD yang terdiri atas dua blok logik. Kondisi eksekusi ON akan dihasilkan jika blok logik atas atau blok logik bawah dalam kondisi ON. Artinya, IR000.00 dan kondisi ON dan IR000.01 dalam kondisi OFF atau IR000.02 dan IR000.03 dalam kondisi ON).



Alamat	Instruksi	Operan
00000	LD	00000
00001	AND NOT	00001
00002	LD	00002
00003	AND	00003
00004	OR LD	---

Gambar 2.13 Contoh penggunaan instruksi blok logika OR LD.

Untuk membuat kode mnemonik diagram tangga yang kompleks, caranya dengan cara membagi membagi diagram tersebut ke dalam blok-blok logik yang besar, kemudian membagi lagi blok yang besar tersebut menjadi blok-blok logik yang lebih kecil, demikian seterusnya hingga tidak perlu lagi dibuat blok yang lebih kecil lagi. Blok-blok ini kemudian masing-masing dikodekan, mulai dari yang kecil, dan digabungkan satu per satu hingga membentuk diagram tangga yang asli. Instruksi blok logik AND LD dan OR LD hanya digunakan untuk menggabungkan dua blok logik saja (blok logik yang digabungkan berupa hasil penggabungan sebelumnya, atau hanya sebuah kondisi tunggal). Gambar 2.14 memperlihatkan suatu contoh diagram tangga yang kompleks, yang dapat dibagi dua blok besar (blok A dan B). Blok A dapat dibagi lagi menjadi dua blok yang lebih kecil (blok A1 dan A2), dan blok B dibagi menjadi dua blok yang lebih kecil, yaitu blok B1 dan B2. Kemudian blok-blok logik yang kecil ini ditulis terlebih dahulu, diawali dengan menuliskan blok A1 (alamat 00000 dan 00001) dan blok A2 (alamat 00002 dan 00003), kemudian digabung menggunakan instruksi blok logik OR LD (alamat 00004). Selanjutnya blok B1 dituliskan (alamat 00005 dan 00006) dilanjutkan dengan blok B2 (alamat 00007 dan 00008) dan digabung dengan instruksi blok logik OR LD (alamat 00009). Hasilnya berupa blok A dan blok B yang kemudian juga digabung menggunakan blok logik AND LD (alamat 00010).



Alamat	Instruksi	Operan
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND	00003
00004	OR LD	—
00005	LD	00004
00006	AND	00005
00007	LD	00006
00008	AND	00007
00009	OR LD	—
00010	AND LD	—
00011	OUT	20003

Blok A1 dan A2

Blok B1 dan B2

Blok A dan B

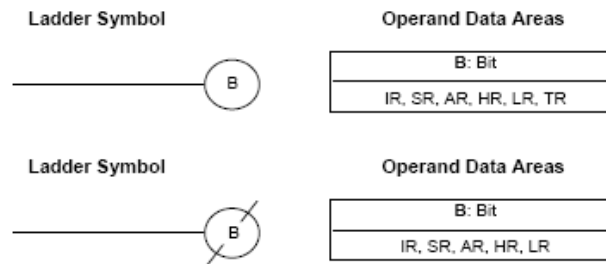
Gambar 2.14 Contoh diagram tangga yang kompleks.

2.7.3 Intruksi Kontrol Bit

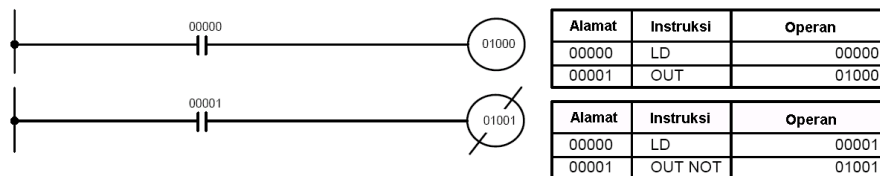
Terdapat intruksui dasar yang dapat digunakan untuk mengontrol status bit secara individual, yaitu DIFFERENTIATE UP (DIFU), DIFFERENTIATE DOWN (DIFD) instruksi ini dituliskan di sisi paling kanan diagram tangga dan membutuhkan sebuah alamat bit sebagai operan. Selain instruksi-instruksi ini digunakan untuk membuat bit-bit keluaran ON atau OFF dalam area IR (ke piranti eksternal), ini juga digunakan untuk mengontrol bit-bit lainnya.

A. Instruksi OUTPUT (OUT) dan OUTPUT NOT (OUT NOT)

Instruksi ini digunakan untuk mengontrol operan yang berkaitan dengan kondisi eksekusi (apakah ON atau OFF). Dengan menggunakan instruksi OUT, maka bit operan akan menjadi ON jika kondisi eksekusinya juga ON, sedangkan OUT NOT akan menyebabkan bit operan menjadi ON jika kondisi eksekusinya OFF. Gambar 2.15 memperlihatkan simbol tangga dan area data operan dari instruksi OUT dan OUT NOT, sedangkan Gambar 2.16 memperlihatkan contoh implementasi kedua instruksi tersebut.



Gambar 2.15 Simbol tangga dan area data operan instruksi OUT dan OUT NOT.

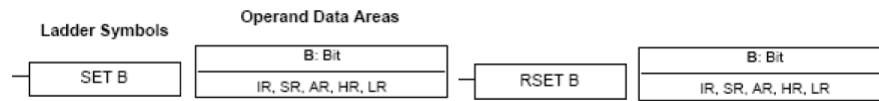


Gambar 2.16 Contoh penggunaan instruksi OUT dan OUT NOT.

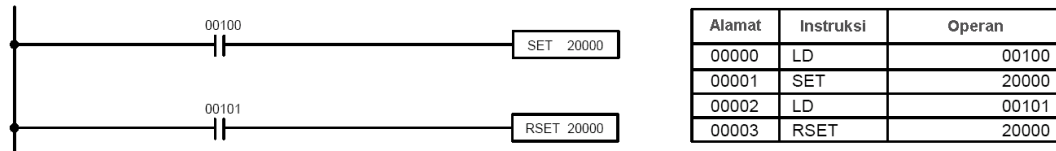
B. SET dan RESET

Instruksi SET dan RESET hampir sama dengan instruksi OUTPUT dan OUTPUT NOT, hanya saja instruksi SET dan RESET ini mengubah kondisi status bit operan saat kondisi eksekusinya ON. Kedua instruksi ini tidak akan mengubah kondisi status bit jika kondisi eksekusinya OFF. Instruksi SET akan meng-ON-kan bit operan saat kondisi eksekusinya ON, tetapi tidak seperti instruksi OUT, SET tidak akan meng-OFF-kan bit operan walaupun kondisi eksekusinya sudah menjadi OFF (setelah ON). Sedangkan instruksi RESET akan meng-OFF-kan bit operan saat kondisi eksekusinya ON, tetapi tidak seperti instruksi OUT NOT, RESET tidak akan meng-ON-kan bit operan walaupun kondisi eksekusinya sudah OFF (setelah ON).

Gambar 2.17 memperlihatkan simbol tangga dan area data operan dari instruksi SET dan RESET, sedangkan Gambar 2.18 memperlihatkan contoh penggunaan kedua instruksi tersebut di atas. Pada diagram tangga tersebut, IR200.00 akan ON jika kondisi IR001.00 ON dan akan terus ON tidak tergantung kondisi IR001.00 selanjutnya, hingga kondisi IR001.01 menjadi ON sehingga me-RESET IR200.00 (menjadi OFF).



Gambar 2.17 Simbol tangga dan area data operan instruksi SET dan RESET.

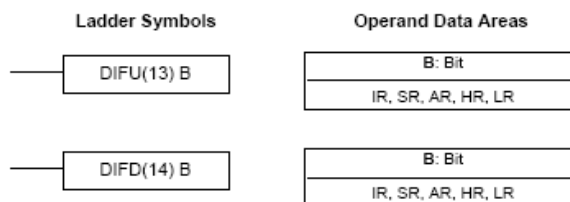


Gambar 2.18 Contoh penggunaan instruksi SET dan RESET.

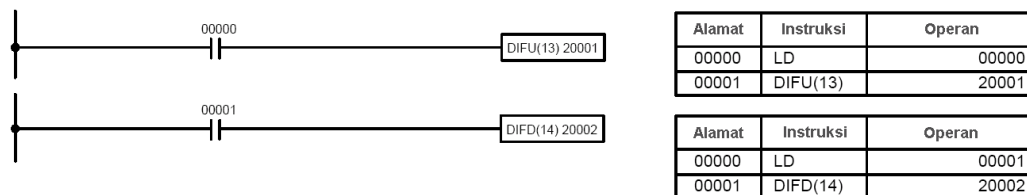
C. DIFFERENTIATE UP (DIFU) dan DIFFERENTIATE DOWN (DIFD)

Terdapat tujuh instruksi dasar yang dapat digunakan untuk mengontrol status bit secara individual, yaitu OUTPUT, OUTPUT NOT, SET, RESET, DIFFERENTIATE UP (DIFU), DIFFERENTIATE DOWN (DIFD), dan KEEP. Semua instruksi ini dituliskan di sisi paling kanan diagram tangga dan membutuhkan sebuah alamat bit sebagai operan. Selain instruksi-instruksi ini digunakan untuk membuat bit-bit keluaran ON atau OFF dalam area IR (ke piranti eksternal), mereka juga digunakan untuk mengontrol bit-bit lainnya.

Gambar 2.19 memperlihatkan simbol tangga dan area data operan dari instruksi DIFU dan DIFD, sedangkan Gambar 2.20 memperlihatkan diagram tangga sederhana yang menggunakan instruksi DIFU dan DIFD. Pada diagram tangga tersebut, IR200.01 akan ON untuk satu siklus (sesaat), jika terjadi transisi perubahan kondisi eksekusi pada IR000.00 dari OFF menjadi ON. Saat DIFU(13) 200.01 dikerjakan lagi untuk siklus berikutnya, IR200.01 tetap akan OFF (tidak tergantung lagi pada status IR000.00). IR200.02 akan ON untuk satu siklus saja jika terjadi transisi perubahan kondisi eksekusi pada IR000.01 dari ON menjadi OFF. Saat DIFD(14)200.02 dikerjakan lagi untuk siklus berikutnya, IR200.02 akan tetap OFF (tidak tergantung lagi status IR000.01).



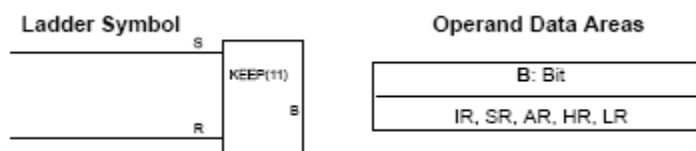
Gambar 2.19 Simbol tangga dan area data operan dari instruksi DIFU dan DIFD.



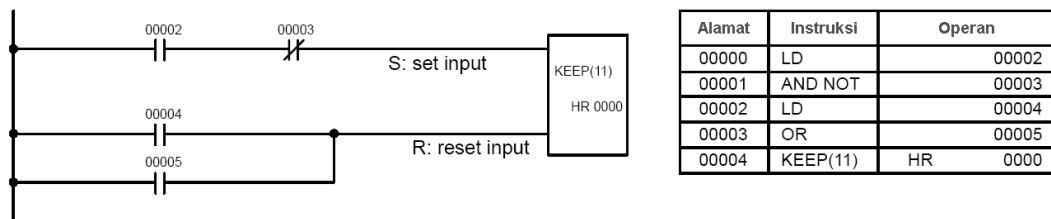
Gambar 2.20 Contoh penggunaan instruksi DIFU dan DIFD.

D. Instruksi KEEP

Instruksi KEEP digunakan untuk menyimpan status suatu bit operan berbasis pada dua kondisi eksekusi. Untuk keperluan ini, instruksi KEEP dihubungkan ke dua garis instruksi. Garis instruksi pertama digunakan untuk meng-ON-kan bit operan, sedangkan garis instruksi kedua digunakan untuk meng-OFF-kan bit operan, hal tersebut akan terjadi jika kondisi eksekusi pada garis instruksi yang terkait ON. Bit operan instruksi KEEP akan dijaga ON atau OFF-nya walaupun ada di dalam bagian diagram yang mengandung INTERLOCK. Gambar 2.21 memperlihatkan simbol tangga dan area data operan dari instruksi KEEP, sedangkan Gambar 2.22 memperlihatkan contoh diagram tangga yang menggunakan instruksi KEEP. HR00.00 akan ON saat IR000.02 dalam kondisi ON dan IR000.03 dalam kondisi OFF, HR00.00 akan tetap ON hingga IR000.04 atau IR000.05 dalam kondisi ON. Karena instruksi ini memerlukan lebih dari satu garis instruksi maka garis-garis instruksi dikodekan terlebih dahulu (alamat 00000 sampai 00003) sebelum menuliskan instruksi KEEP-nya (alamat 00004).



Gambar 2.21 Simbol tangga dan area data operan dari instruksi KEEP.

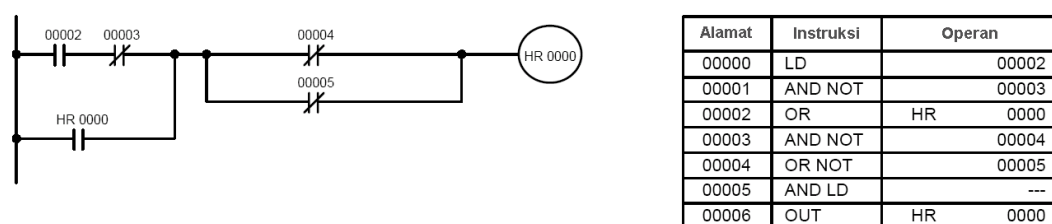


Gambar 2.22 Contoh penggunaan instruksi KEEP.

Dalam merancang suatu program diagram tangga, untuk membuat program lebih efektif, ada dua hal yang perlu diperhatikan dan dapat dimanfaatkan, yaitu:

- **Bit-bit Penyimpan (*Self-maintained*)**

Walaupun instruksi KEEP dapat digunakan untuk membuat bit-bit yang bisa menahan sendiri, kadangkala dibutuhkan membuat bit-bit yang bisa menahan sendiri dengan cara yang lain, sehingga bit-bit tersebut bisa di-OFF-kan pada bagian program yang mengandung INTERLOCK. Untuk membuat bit-bit tersebut, bit operan suatu instruksi OUT dibuat sebagai suatu kondisi untuk OUT yang sama dalam suatu bentuk penggabungan OR, sehingga bit operan OUT akan tetap ON atau OFF hingga muncul perubahan kondisi pada bit-bit yang lain. Sebagai contoh, kita bisa mengubah diagram tangga pada Gambar 2.22 (yang menggunakan instruksi KEEP) dengan menghilangkan instruksi KEEP tersebut dan meletakkan bit operan OUT ke suatu garis instruksi tersendiri yang kemudian digabung dengan OR. Hasilnya ditunjukkan pada Gambar 2.23



Gambar 2.23 Contoh diagram tangga yang menggunakan bit penyimpan.

- **Bit-bit Kerja (*Work Bits*) atau Relai Internal**

Dalam pemrograman diagram tangga, mengkombinasikan kondisi-kondisi untuk secara langsung menghasilkan kondisi eksekusi seringkali sulit dilakukan. Kesulitan ini dapat ditangani dengan cara menggunakan beberapa bit untuk memicu instruksi lain secara tidak langsung. Bit-bit I/O dan bit-bit terdedikasi lainnya tidak dapat digunakan sebagai bit-bit kerja. Semua bit-bit yang ada di

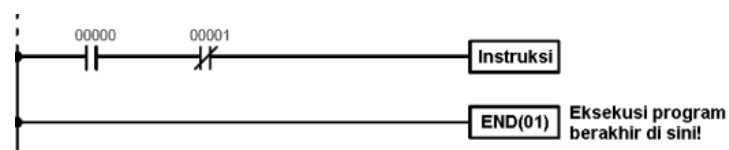


dalam area IR yang tidak dialokasikan untuk I/O dan beberapa bit yang tidak digunakan dalam area AR, dapat digunakan sebagai bit-bit kerja. Bit-bit kerja biasanya sering digunakan bersama-sama dengan instruksi-instruksi OUT, OUT NOT, DIFU, DIFD, dan KEEP. Bit kerja pertama kali langsung digunakan oleh instruksi-instruksi tersebut sebagai bit operan sehingga kemudian bisa digunakan sebagai kondisi yang akan menentukan instruksi-instruksi lainnya. Bit-bit kerja dapat digunakan untuk menyederhanakan diagram tangga pada saat suatu kombinasi kondisi perlu diulang-ulang di beberapa bagian dalam diagram tangga yang bersangkutan.

2.7.4 Instruksi END

Instruksi END merupakan instruksi terakhir yang harus dituliskan atau digambarkan dalam diagram tangga. CPU pada PLC akan mengerjakan semua instruksi dalam program dari awal (baris pertama) sampai ditemui instruksi END yang pertama, sebelum kembali lagi mengerjakan instruksi dalam program dari awal (artinya instruksi-instruksi yang ada di bawah instruksi END akan diabaikan). Instruksi END tidak memerlukan operan dan tidak boleh diawali dengan suatu kondisi seperti pada instruksi lainnya.

Suatu diagram tangga atau program PLC harus diakhiri dengan instruksi END, jika tidak maka program tidak dijalankan sama sekali. Angka yang dituliskan pada instruksi END pada kode mnemonik merupakan kode fungsinya. Gambar 2.24 memperlihatkan contoh penggunaan instruksi END.



Alamat	Instruksi	Operan
00500	LD	00000
00501	AND NOT	00001
00502	Instruction	
00503	END(01)	---

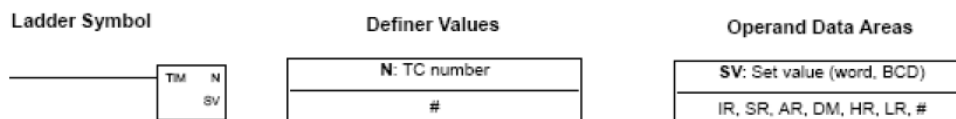
Gambar 2.24 Contoh penggunaan instruksi END.



2.7.5 Instruksi TIMER dan COUNTER

a. Instruksi TIMER (TIM)

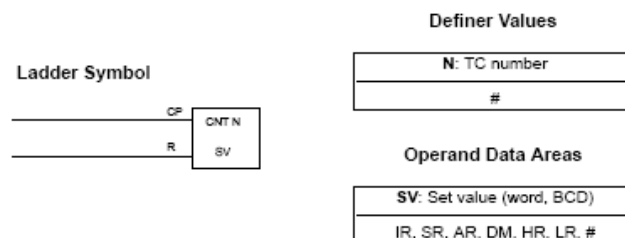
Instruksi TIM dapat digunakan sebagai timer (pewaktu) ON-delay pada rangkaian relai. Gambar 2.25 memperlihatkan simbol tangga dan area data operan dari instruksi TIM. Instruksi TIM membutuhkan angka timer (N), dan nilai set (SV) antara 0000 sampai 9999 (artinya 000,0 sampai 999,9 detik).



Gambar 2.25 Simbol tangga dan area data operan dari instruksi TIMER (TIM).

b. Instruksi COUNTER (CNT)

CNT yang digunakan di sini adalah counter penurunan yang diset awal. Penurunan satu hitungan setiap kali saat sebuah sinyal berubah dari OFF ke ON. Counter harus diprogram dengan input hitung, input reset, angka counter, dan nilai set (SV) Nilai set ini adalah 0000 sampai 9999. Gambar 2.26 memperlihatkan simbol tangga dan area data operan dari instruksi CNT.

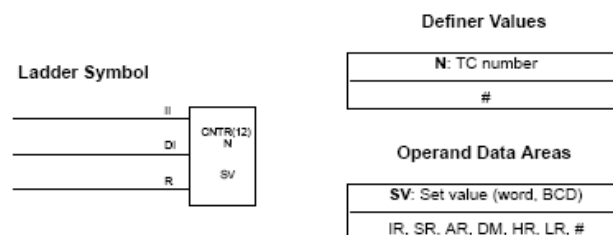


Gambar 2.26 Simbol tangga dan area data operan dari instruksi COUNTER (CNT).

Instruksi untuk *Reversible Counter* adalah CNTR. CNTR(12) adalah *reversible, up/down circular counter*, sehingga bisa dipakai untuk menghitung antara nol dan SV [set value] tergantung dari dua masukan, apakah *increment* pada input (II) atau *decrement* pada input (DI). Nilai saat itu (PV) akan naik satu jika input II mendapat masukan ON kemudian OFF, dan akan turun satu jika input DI mendapat masukan ON kemudian OFF. Jika keduanya mendapatkan masukan bersama maka nilai saat itu tidak berubah. Jika nilai saat itu 0000 dan mendapat



masukannya turun maka nilai saat itu akan berubah ke nilai set value SV dan *Completion Flag* akan ON, jika nilai saat itu melebihi SV maka PV akan menjadi 0000. CNTR(12) akan reset [nilai saat itu menjadi nol] jika input reset R berubah dari OFF ke ON. PV tidak akan naik/turun selama R ON kemudian akan menghitung lagi jika R OFF. nilai PV untuk CNTR(12) tidak akan reset pada saat program interlock atau oleh karena gangguan supply. Gambar 2.27 memperlihatkan simbol tangga dan area data operan dari instruksi CNTR.



Gambar 2.27 Simbol tangga dan area data operan dari instruksi CNTR.

2.8 Jenis-jenis Alat Pemrogram

Ada beberapa jenis alat untuk memasukkan program ke dalam PLC yaitu komputer yang dilengkapi dengan software ladder misalnya CX-Programmer, Siswin 3.4 dan Konsol Pemrogram.

Dengan software ladder CX-Programmer, program yang dimasukkan ke dalam PLC dapat berbentuk diagram ladder atau kode mneumonik, tetapi Konsol Pemrogram hanya dapat memasukkan program dalam bentuk kode mneumonik

2.9 Mode Operasi PLC^[8]

Operasi PLC dikategorikan dalam tiga mode yaitu : PROGRAM, MONITOR, dan RUN. Pilihan mode operasi harus dipilih dengan tepat sesuai dengan aktifitas dalam sistem kendali PLC.

1. Mode PROGRAM digunakan untuk membuat dan mengedit program, menghapus memori, atau mengecek kesalahan program. Pada mode ini, program tidak dapat dieksekusi/ dijalankan.



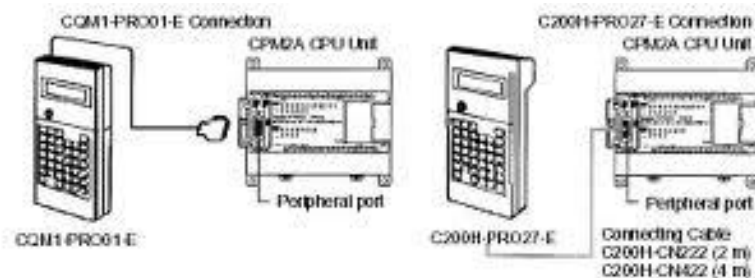
2. Mode MONITOR digunakan menguji operasi sistem, seperti memonitor status operasi, melaksanakan instruksi force set dan force reset bit I/O, merubah SV (Set Value) dan PV (Present Value) timer dan counter, merubah data kata, dan mengedit program online.
3. Mode RUN digunakan untuk menjalankan program. Status operasi PLC dapat dimonitor dari peralatan pemrogram, tetapi bit tdk dapat di paksa set/ reset dan SV/PV timer dan counter tidak dapat diubah.

2.10 Sambungan Alat Pemrogram^[8]

PLC dapat disambung ke Konsol Pemrogram atau komputer dengan software ladder seperti CX-Programmer, SSS (Sysmac Support Software) atau Syswin dan Programmable Terminal.

1. Sambungan Konsol Pemrogram

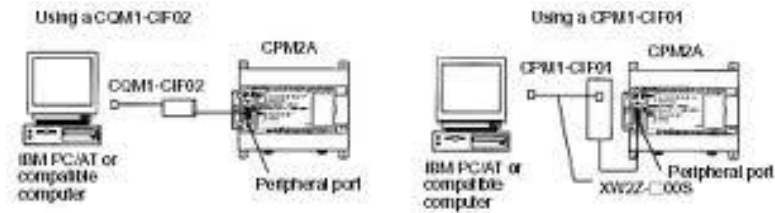
Hubungkan Konsol Pemrogram ke port peripheral PLC. Konsol Pemrogram tidak dapat disambung ke port RS-232C. PLC akan otomatis berkomunikasi dengan Konsol Pemrogram tanpa memandang metode komunikasi yang dipilih pada saklar komunikasi PLC.



Gambar 2.28 Sambungan Konsol Pemrograman.

2. Sambungan Komunikasi Host Link

Komunikasi Host Link adalah komunikasi antara PLC dan komputer yang didalamnya diinstal software ladder. Komputer dapat disambung ke port peripheral atau port RS-232C PLC. Port peripheral dapat beroperasi dalam mode Host Link atau mode peripheral bus. Port RS-232C beroperasi hanya dalam mode Host Link Komputer dapat disambung ke port peripheral PLC dengan adapter RS-232C : CQM1-CIF02 atau CPM1-CIF01.



Gambar 2.29 Sambungan Komunikasi Host Link.

3. Sambungan Komunikasi NT Link

Komunikasi NT Link adalah komunikasi antara PLC dan Programmable Terminal. Pada Link NT 1:1, PLC dapat disambung langsung ke Programmable Terminal yang disambung ke port RS-232C. Ia tidak dapat disambung ke port peripheral.



Gambar 2.30 Sambungan Komunikasi NT Link.

2.11 Garbarata^[9]

Garbarata merupakan salah satu peralatan yang berperan penting dalam menunjang pelayanan jasa angkutan udara dan merupakan suatu alat yang tepat untuk menurunkan dan menaikkan penumpang pesawat guna melindungi gangguan dari hujan, angin, debu dan tiupan mesin jet serta sebagai pemisah antara penumpang dengan petugas di darat. Garbarata digunakan untuk menghubungkan gedung terminal ke pesawat terbang dan unit ini terdiri dari tunnel-tunnel atau lorong-lorong yang saling berhubungan satu dengan yang lainnya dan dapat dijalankan dari tempat parkir gedung terminal ke pesawat udara. Pergerakan Garbarata tersebut dikemudikan oleh operator AMC (Apron Move Control) melalui pengendali listrik yang dipasang pada panel kemudi di dalam



cabin. Perlu diketahui bahwa semua dari pergerakan mekanis dari garbarata terpusat pada alat pengendali yang bernama PLC yang berfungsi sebagai pengendali pergerakan dan sebagai pengendali pembatas kerja dari pergerakan Garbarata itu sendiri.



Gambar 2.31 Garbarata^[10]

2.11.1 Gerakan-gerakan utama pada garbarata yang dikendalikan PLC^[9]

1. Wheel Bogie (Gerakan horisontal) yaitu gerakan maju mundur, membelok ke kiri dan ke kanan, untuk mendekat dan menjauhi pesawat terbang.
2. Gerakan Vertikal (Lift Column), yaitu gerakan naik turun garbarata.
3. Cabin rotasi yaitu gerakan berputarnya kabin untuk mensejajarkan posisi dengan pesawat terbang.
4. Canopy atau closure yaitu gerakan penutupnya canopy pada saat docking telah selesai.
5. Lantai Cabin yaitu pergerakan naik dan turunya lantai bagian depan garbarata yang terletak pada bagian depan cabin.



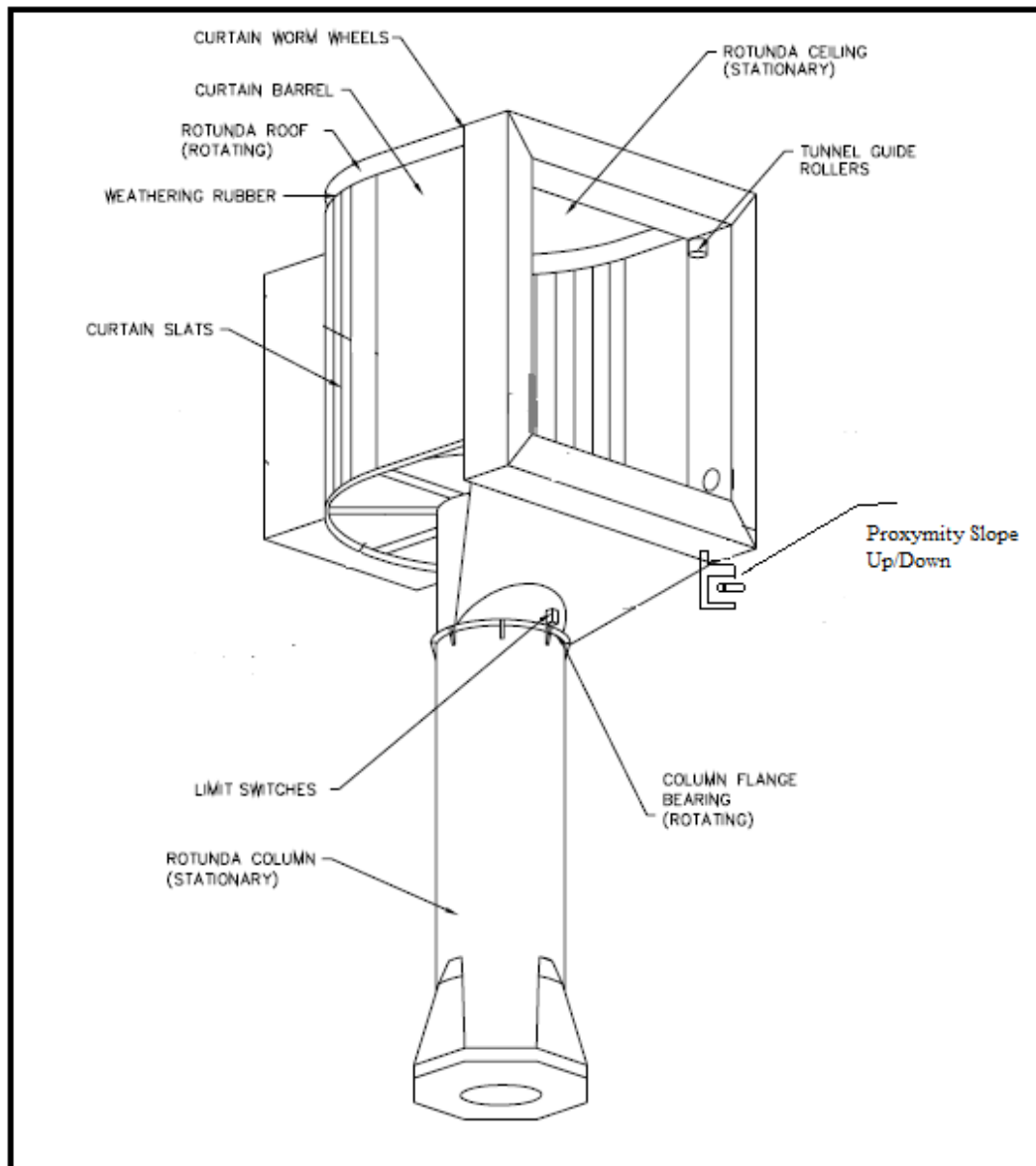
2.12 Bagian-bagian konstruksi mekanis utama dari garbarata^[9]

Garbarata adalah seluruh komponen peralatan yang terpasang, adapun bagian – bagian utama Mekanis yang terdapat di garbarata itu yang di antaranya :

1. Rotunda
2. Tunnel (lorong)
3. Vertical lift coulumn
4. Wheel Bogie (roda)
5. Cabin

2.12.1 Rotunda

Rotunda merupakan poros penghubung gedung terminal dengan lorong (Tunnel). Pada pergerakannya rotunda dapat berrotasi dengan pergerakan dari roda. Pergerakan rotasi garbarata dengan rotunda sebagai porosnya dilengkapi dengan pengaman dua Limit Switch yang dapat di lihat pada gambar 2.32 Limit Switch dipasang sebagai pembatas dari kerja pergerakan Full Rotasi (Sterring) dan pemberi signal alarm full sterring pada pergerakan Rotasi Rotunda. Pada Rotunda juga terpasang Proxymity Sensor sebagai pendeteksi batasan ketinggian maksimum dan minimum pada tunnel saat pergerakan memendek agar saat garbarata memposisikan ke area parkirnya garbarata harus pada posisi yang standar. Apabila posisi ketinggian garbarata maksimum ataupun minimum pada saat garbarata melakukan pergerakan mundur (memendek) ke posisi parkirnya, proxymity yang terpasang pada ujung bagian tunnel akan mendeteksi Camp pada bagian rotunda dan memberikan alarm Garbarata Slope, selanjutnya proximity memutus pergerakan mundurnya garbarata. Pergerakan dapat dilakukan apabila proxymity tidak lagi terdeteksi yang berarti garbarata dalam posisi ketinggian yang standar untuk ke posisi parkirnya. Proxymity dipasang sebagai pendeteksi pengaturan ketinggian pada tunnel pada saat tunnel garbarata melakukan peggerakan memendek (Extend) agar posisi bagian cabin garbarata tidak terlalu condong ke bawah/atas.

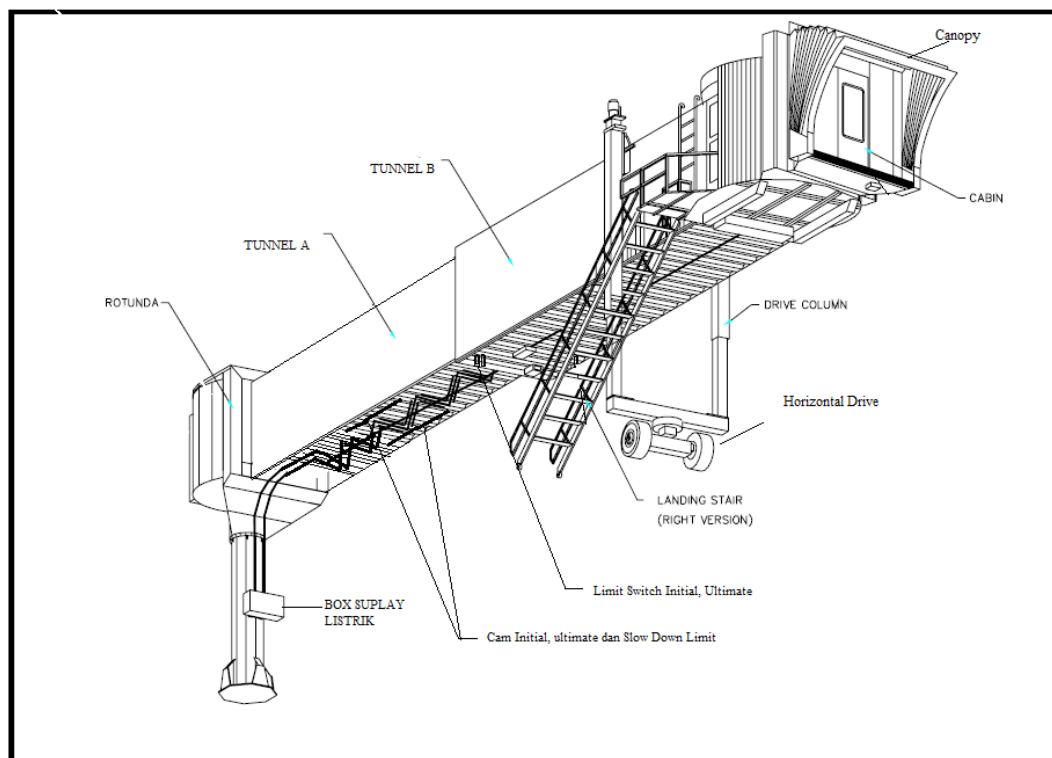
Gambar 2.32 Rotunda.^[11]

2.12.2 Tunnel (lorong)

Tunnel adalah bagian mekanis garbarata yang menghubungkan bagian Rotunda ke bagian Cabin pada garbarata. Pada Garbarata terdapat dua buah tunnel seperti pada gambar 2.33 terlihat ada Tunnel A yang terhubung ke bagian Rotunda dan Tunnel B yang terhubung ke bagian Cabin dari Garbarata. Tunnel adalah salah satu pergerakan mekanis dari garbarata yang dapat memanjang (Retract) dan memendek (Extend). Pergerakan memanjang untuk memposisikan



Garbarata pada bagian pintu pesawat, sedangkan memendek adalah pergerakan garbarata setelah selesai docking (memasukan penumpang ke pesawat) untuk memposisikan garbarata ke posisi parkirnya. Pada pergerakan memanjang ataupun memendek Garbarata dilengkapi 3 buah Limit Switch yang terpasang pada bagian bawah tunnel dilihat pada Gambar 2.33 Limit Switch pertama berfungsi sebagai pemberi signal pengaturan Slow Down (Memperlambat pergerakan maju/mundur) pada Garbarata. Sedangkan limit switch kedua berfungsi sebagai alarm full memanjang/memendek dan limit switch ketiga berfungsi sebagai memberhentikan dari pergerakan maju/ mundur pada whell bogie / roda.



Gambar 2.33 Tunnel (lorong)^[10]

2.12.3 Vertical lift column

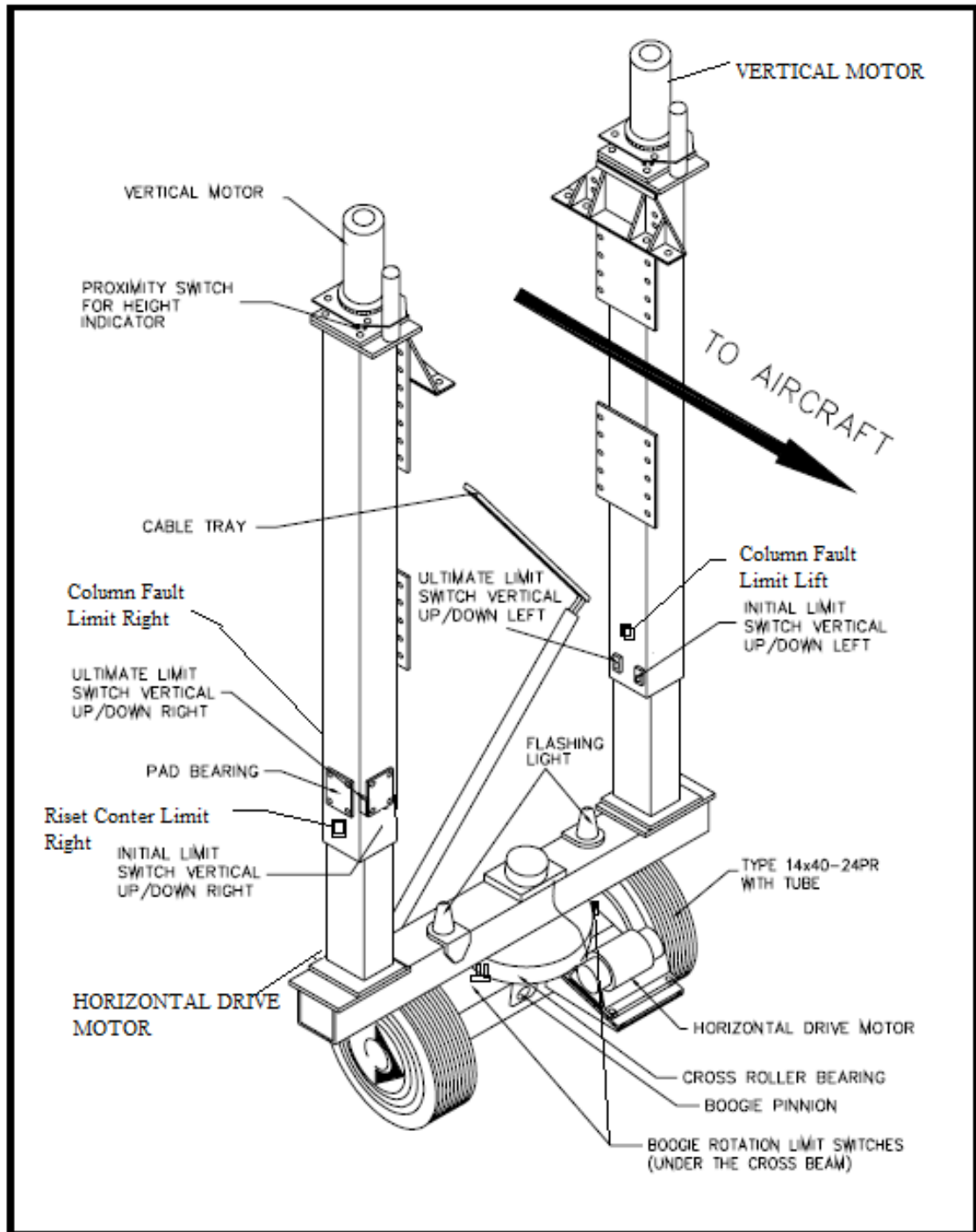
Pada vertical lift column atau tiang penyangga untuk naik turunnya garbarata ini terdapat komponen untuk gerakan-gerakan vertical up maupun



down. Vertical lift column penempatannya menyatu dengan bagian Roda (wheel bogie). Vertical lift column terdiri dari bagian-bagian tabung dari ball nut yang berada dalam tabung dan unit motor untuk penggerak vertical up/down. Pada vertical lift column ini terdapat juga pengaman limit switch sebagai pengaman dari batasan full maksimum up/down, perbedaan ketinggian dari vertical lift column kiri/kanan dan terdapat juga proximity sensor yang berfungsi menghitung ketinggian dari garbarata dari putaran motor vertical dengan perbandingan rasio 1:100 putaran motor seperti terlihat pada Gambar 3.34. Pada vertical lift column ini terdapat dua buah motor induksi 3 fase yang terletak di kedua sisi tiang dimana kerja dari kedua motor induksi tersebut bekerja serempak dengan prinsip dua arah putaran untuk intruksi dari kerja naik (up) atau turunnya(down) Garbarata.

2.12.4 Wheel bogie (roda)

Wheel bogie adalah salah satu bagian utama dari garbarata yang terletak di bagian bawah garbarat sebagai panggerak utama garbarata (roda) yang dilengkapi dengan dua buah motor listrik yang bekerja dengan dua arah putaran untuk pengaturan maju, mundur ataupun berbeloknya garbarata terlihat pada gambar 3.24. Selain motor listrik wheel bogie juga dilengkapi dengan limit switch dan juga inverter sebagai pengaturan kecepatan motor dan pembatas pergerakan steering (berbelok) saat roda garbarata telah mendekati batasan limit yang telah di setting ke camp yang ada di garbarata saat roda berbelok ke kiri/ kekanan. Sedangkan limit switch yang terpasang pada Wheel bogie terdapat empat buah yang berfungsi sebagai batas kerja initial dan ultimate/maksimum dari pergerakan berbelok kiri dan kanan terlihat pada gambar 3.24 selain empat limit switch diatas sebagai pembatas kerja roda, juga terdapat limit switch dan sensor lain yang interlock dengan pergerakan dari roda yaitu limit ultimate tunnel, full steering pada rotunda, dan bumper limit pada cabin garbarata.



Gambar 2.34 Vertical Lift Column dan whell Bogie.^[11]

2.12.5 Cabin

Pada cabin terdapat tiga pergerakan yang diantaranya cabin rotasi (rotasi kiri/kanan), canopy dan lantai cabin terlihat pada gambar 2.36. Yang mana setiap



pergerakan tersebut terdapat perangkat pengaman berupa limit switch atau proximity detector sebagai pembatas kerja dari masing-masing pergerakan.

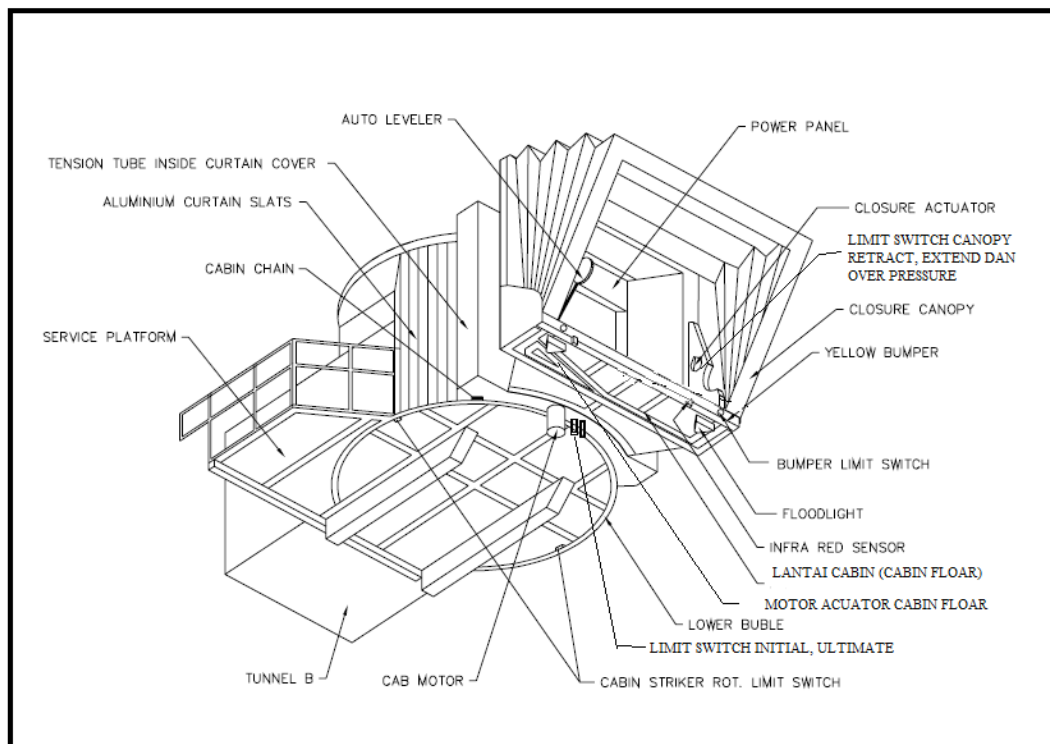
Pertama, pergerakan cabin rotasi adalah pergerakan cabin berrotasi ke kiri/ke kanan untuk menyesuaikan bagian cabin itu sendiri ke bagian pintu pesawat terlihat seperti gambar 2.36. Pada pergerakan rotasi cabin terdapat 2 buah limit switch yang berfungsi sebagai alarm full rotasi (initial) kiri/kanan dan pemutus kerja dari motor (ultimate) kiri/kanan garbarata. Dari pergerakan cabin rotasi terdapat satu buah motor listrik sebagai penggerak dari rotasi cabin.

Kedua, pergerakan Canopy adalah pergerakan menutupnya bagian Closure (tudung) dari garbarata. Pada canopy pergerakannya terbagi menjadi dua bagian yaitu canopy kiri dan kanan yang masing-masing canopy kiri/kanan di gerakan oleh tombol tekan up dan down. Pada canopy pergerakan ya dilengkapi 3 buah limit switch yang terpasang di kedua bagian canopy kiri dan kanan. Limit switch pertama berfungsi sebagai pembatas dari kerja full menutup (Retract), limit switch kedua berfungsi sebagai full membuka (Extend) dan limit switch ketiga berfungsi sebagai over pressure dan sekaligus memberikan alarm dari kerja canopy. Dari pergerakan canopy terdapat dua buah motor listrik Acuator DC terlihat pada gambar 2.35 yang terpasang di kedua sisi canopy kiri dan kanan sebagai penggerak dari canopy.

Ketiga, pergerakan Lantai cabin adalah pergerakan naik dan turun dari lantai cabin untuk menyesuaikan posisi dari lantai cabin. Pada pergerakan lantai cabin naik(up)maupun turun (down) dilengkapi pengaman limit switch yang berfungsi sebagai pembatas dari kerja up/down dari lantai cabin. Dari pergerakannya lantai cabin terdapat satu buah motor listrik Acuator DC terlihat pada gambar 2.35 yang terpasang pada bagian bawah lantai cabin sebagai penggerak dari lantai cabin.

Gambar 3.35 Motor Acuator.^[12]

Selain itu pada cabin juga terdapat bumper limit yang berfungsi memberhentikan kerja dari pergerakan maju pada roda (Wheel bogie) saat bumper limit telah terdeteksi oleh badan pesawat saat cabin garbarata menyentuh badan pesawat.

Gambar 2.36 Cabin, Canopy dan Lantai Cabin.^[11]