

**RANCANG BANGUN PANGGILAN OTOMATIS
KERETA LISTRIK (*TRAM*) PADA STASIUN
MENGUNAKAN *PROGRAMMABLE LOGIC
CONTROLLER MASTER-K120S***

TUGAS AKHIR

Oleh

DADANG ISNANDAR

06 06 04 238 0



**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

**RANCANG BANGUN PANGGILAN OTOMATIS
KERETA LISTRIK (*TRAM*) PADA STASIUN
MENGUNAKAN *PROGRAMMABLE LOGIC
CONTROLLER MASTER-K120S***

Oleh

DADANG ISNANDAR

06 06 04 238 0



TUGAS AKHIR INI DIAJUKAN UNTUK MELENGKAPI
SEBAGIAN PERSYARATAN MENJADI SARJANA TEKNIK

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul :

**RANCANG BANGUN PANGGILAN OTOMATIS
KERETA LISTRIK (*TRAM*) PADA STASIUN MENGGUNAKAN
*PROGRAMMABLE LOGIC CONTROLLER MASTER-K120S***

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari tugas akhir yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, Juni 2008

Dadang Isnandar

NPM. 0606042380

PENGESAHAN

Tugas akhir dengan judul :

**RANCANG BANGUN PANGGILAN OTOMATIS
KERETA LISTRIK (*TRAM*) PADA STASIUN MENGGUNAKAN
*PROGRAMMABLE LOGIC CONTROLLER MASTER-K120S***

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang ujian tugas akhir.

Depok, Juni 2008
Dosen Pembimbing,

Budi Sudiarto, S.T, M.T

NIP. 040 705 0181

UCAPAN TERIMAKASIH

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan nikmat serta karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini.

Dalam kesempatan ini, penulis ingin mengucapkan terima kasih khususnya kepada Bapak **Budi Sudiarto, S.T, M.T** selaku pembimbing yang telah bersedia meluangkan waktunya untuk memberikan gagasan, konsultasi, petunjuk, saran-saran, dan motivasi serta kemudahan lainnya sehingga tugas akhir ini dapat diselesaikan dengan baik.

Selain itu juga penulis ingin mengucapkan ucapan terima kasih kepada :

1. Bapak Dr. Ir. Arman Djohan Diponegoro, M.Eng
2. Kedua orang tua serta adik-adik yang telah memberikan bantuan doa serta dukungannya
3. Rekan-rekan Eks06, Ady, Ingot, Mutaqien, Rijal, Taufik, Budi, Vebby, Aulia (al) atas kebersamaanya
4. Noel gallagher atas motivasinya

Dan semua pihak lain yang telah membantu penyelesaian tugas akhir ini.

Dadang Isnandar
NPM 06 06 04 238 0
Departemen Teknik Elektro

Dosen Pembimbing
Budi Sudiarto, S.T, M.T

**RANCANG BANGUN PANGGILAN OTOMATIS
KERETA LISTRIK (*TRAM*) PADA STASIUN MENGGUNAKAN
PROGRAMMABLE LOGIC CONTROLLER MASTER-K120S**

ABSTRAK

Kereta listrik (*tram*) merupakan salah satu *Mass Rapid Transportation* (MRT) yang akan diaplikasikan di lingkungan kampus Universitas Indonesia. *Tram* menggunakan energi listrik sebagai sumber tenaga, hal itu menjadikan *tram* ramah terhadap lingkungan. Sumber tenaga listrik dapat dikombinasikan dengan peralatan kontrol elektronik, seperti *electric drive* serta *Programmable Logic Controller* (PLC). Berkaitan dengan hal tersebut, maka dalam penulisan tugas akhir ini akan dilakukan rancang bangun miniatur *tram* yang merupakan tahap awal untuk mendapatkan sistem transportasi menggunakan kereta listrik.

Rancang bangun yang dilakukan adalah mendapatkan program *ladder* untuk mengendalikan panggilan dan pemberhentian *tram* pada stasiun dengan menggunakan PLC serta mengaplikasikannya kedalam miniatur *tram*. Miniatur terbagi atas tiga buah stasiun yang terdapat dalam sistem, dimana masing-masing stasiun terdiri dari beberapa sensor dan tombol tekan yang digunakan sebagai Input PLC. Pada tugas akhir ini telah berhasil dirancang dan dibangun serta dianalisis panggilan otomatis *tram* dari stasiun secara utuh, sehingga pada tahap selanjutnya dapat membantu kesempurnaan dalam pembuatan sistem transportasi menggunakan *tram* yang akan diaplikasikan di dalam kampus Universitas Indonesia.

Kata kunci : *Tram, Programmable Logic Controller, Ladder, Miniatur*

Dadang Isnandar
NPM 06 06 04 238 0
Electrical Departemen Engineering

Counsellor
Budi Sudiarto, S.T, M.T

**DESIGN AND BUILD AUTOMATIC CALL OF LIGHT RAIL TRAIN
(TRAM) AT STATION USING PROGRAMMABLE LOGIC
CONTROLLER MASTER-K120S**

ABSTRACT

A Light Rail Train (tram) is a type of Mass Rapid Transportation (MRT) which to be applied in University of Indonesia. Tram has used electric energy as power source and it will become friendly for environment. An electric power source can combine with Programmable Logic Controller (PLC) and electric drive. This final project will be done by design and build a miniature of tram. It is an early stage to get a system of transportation using a tram.

Main purpose the project is to get a ladder program for controlling call and stop tram at station with PLC and also applicated into miniature of tram. A tram miniature consists of three station in the system, where each station have a number of push-button and censor (input in PLC). This final project have success to analyze, build and design an automatic call for tram. For the next stage, the project can help engineer making a full-system of electric train (tram) which latter will be applied in University of Indonesia.

Keywords : Tram, Programmable Logic Controller, Ladder, Miniature

DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN TUGAS AKHIR	iii
PENGESAHAN	iv
UCAPAN TERIMA KASIH	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR SINGKATAN	xiii
DAFTAR ISTILAH	xiv
BAB 1 PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 PERUMUSAN MASALAH	2
1.3 TUJUAN PENULISAN	2
1.4 BATASAN MASALAH	2
1.5 METODOLOGI PENELITIAN	3
1.6 SISTEMATIKA PENULISAN	3
BAB 2 <i>CONTROLLER</i> PADA KERETA LISTRIK (<i>TRAM</i>)	4
2.1 <i>PROGRAMMABLE LOGIC CONTROLLER</i> (PLC)	5
2.1.1 Bagian-bagian PLC	7
2.1.2 Prinsip Kerja	8
2.1.3 Metode Memprogram PLC	9
2.1.4 Perancangan Program PLC	10
2.2 PEMROGRAMAN PLC	10
2.2.1 Instruksi PLC Master-K120S	11
2.2.2 Metode Y-Map	12
BAB 3 RANCANG BANGUN	18
3.1 <i>LAY-OUT</i>	18
3.2 DIAGRAM ALIR	21

3.3	DESKRIPSI KERJA <i>TRAM</i>	23
3.4	ALGORITMA PROGRAM	25
3.5	BLOK DIAGRAM	25
BAB 4	ANALISA DAN UJICOBA PROGRAM	27
4.1	PROGRAM <i>LADDER</i> SISTEM PANGGILAN OTOMATIS <i>TRAM</i>	28
4.2	ANALISA PROGRAM	28
4.2.1	Program Kontrol <i>Input</i>	28
4.2.2	Program Kontrol Utama	30
4.2.3	Program Kontrol <i>Output</i>	34
4.3	UJICOBA PROGRAM	35
4.3.1	<i>Tram</i> di Stasiun1 (Skenario Pertama)	36
4.3.2	<i>Tram</i> di Stasiun2 (Skenario Kedua)	41
4.3.3	<i>Tram</i> di Stasiun3 (Skenario Ketiga)	45
BAB 5	KESIMPULAN	49
	DAFTAR ACUAN	50
	DAFTAR PUSTAKA	51
	LAMPIRAN	52

DAFTAR GAMBAR

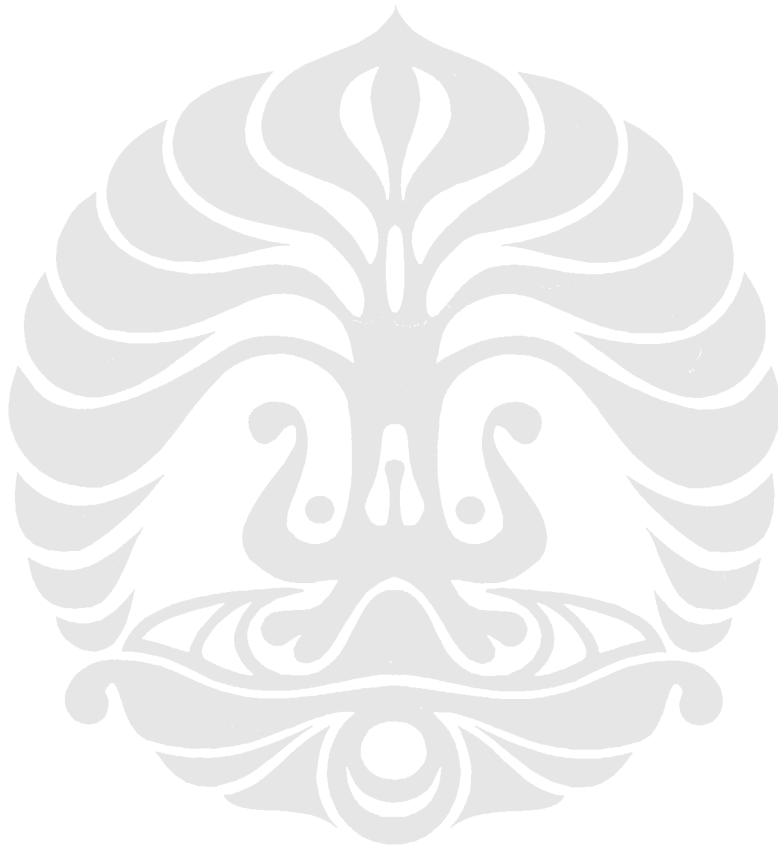
	Halaman
Gambar 2.1	Contoh bentuk fisik <i>tram</i> 4
Gambar 2.2	Contoh bentuk fisik PLC “LG Master-K 120S 30 I/O” 7
Gambar 2.3	Blok diagram dasar sistem kontrol PLC 9
Gambar 2.4	Blok diagram lengkap sistem kontrol PLC 9
Gambar 2.5	Bagian-bagian program untuk <i>timer</i> 11
Gambar 2.6	<i>Timer</i> TON dengan <i>setting value</i> 10 detik 12
Gambar 2.7	<i>Self-holding relay</i> pada koil M0000 13
Gambar 2.8	Program utama kontrol proses metode Y-map SHR 15
Gambar 2.9	<i>Output</i> tipe (a) 16
Gambar 2.10	<i>Output</i> tipe (b) 16
Gambar 2.11	<i>Output</i> tipe (c) 17
Gambar 2.12	<i>Output</i> tipe (d) 17
Gambar 3.1	<i>Lay-out plant tram</i> 19
Gambar 3.2	<i>Wiring diagram</i> miniatur <i>tram</i> 21
Gambar 3.3	Miniatur <i>tram</i> 21
Gambar 3.4	Diagram alir sistem panggilan otomatis <i>tram</i> 22
Gambar 3.5	Blok Diagram Sistem Panggilan <i>Tram</i> 25
Gambar 3.6	Tombol yang terdapat pada stasiun 26
Gambar 4.1	Program kontrol panggilan <i>Push-button</i> 29
Gambar 4.2	Program kontrol menghentikan <i>tram</i> 29
Gambar 4.3	Program kontrol <i>tram</i> di stasiun1 30
Gambar 4.4	Program kontrol <i>tram</i> di stasiun2 32
Gambar 4.5	Program kontrol <i>tram</i> di stasiun3 33
Gambar 4.6	Program me- <i>reset</i> atau melepaskan penguncian SHR 34
Gambar 4.7	Program kontrol <i>output</i> 34
Gambar 4.8	Kondisi awal program di stasiun1 35
Gambar 4.9	Kondisi terdapat panggilan dari stasiun2 36
Gambar 4.10	Kondisi <i>tram</i> berhenti di stasiun2 37

Gambar 4.11	Kondisi <i>tram</i> berhenti di stasiun2 selama 10 detik	38
Gambar 4.12	Kondisi <i>tram</i> menuju stasiun3	39
Gambar 4.13	Kondisi program dimana <i>tram</i> berada di stasiun3	39
Gambar 4.14	Kondisi pemberhentian akhir <i>tram</i> di stasiun3	40
Gambar 4.15	Kondisi program dimana tombol STOP aktif	41
Gambar 4.16	Kondisi program dimana tombol START aktif	43
Gambar 4.17	Kondisi menuju stasiun1 dari stasiun3 (<i>reset</i> program)	44
Gambar 4.18	Kondisi <i>tram</i> menuju stasiun1 dari stasiun3	45
Gambar 4.19	Kondisi di stasiun1 setelah program <i>reset</i> aktif	47



DAFTAR TABEL

	Halaman
Tabel 2.1 Deskripsi kerja Y-map <i>self-holding relay</i> dengan 5 <i>output</i>	13
Tabel 2.2 Tabel macam-macam tipe <i>output</i> dari metode Y-map SHR	14
Tabel 3.1 <i>Input</i> dan <i>Output</i> PLC	20



DAFTAR SINGKATAN

CPU	Central Processing Unit
DC	Direct Current
EPROM	Erasable Programmable Read-Only Memory
I/O	Input / Output
kB	kilo Byte
LED	Light Emmitting Diode
mA	mili ampere
MRT	Mass Rapid Transportation
NC	Normally Close
NO	Normally Open
PLC	Programmable Logic Controller
PROM	Programmable Read Only Memory
SHR	Self Holding Relay

DAFTAR ISTILAH

Console. alat yang digunakan untuk memprogram PLC dengan metode offline.

KGLWIN. software yang digunakan dalam pemrograman PLC tipe MASTER-K.

Ladder. merupakan salah satu metode dalam pemrograman PLC.

Tram. suatu sistem kereta api penumpang yang beroperasi di kawasan perkotaan, konstruksinya ringan dan dapat berjalan bersama lalu lintas lain atau dalam lintasan khusus

Normally Open. Switch dengan kondisi awal terbuka.

Normally Close. Switch dengan kondisi awal tertutup.

Relay. merupakan rangkaian yang bersifat elektronis sederhana dan tersusun oleh medan elektromagnet (kawat koil) dan kontak-kontak atau saklar.

Self Holding Relay (SHR). teknik dalam pemrograman PLC yang memiliki arti penguncian terhadap diri koil *relay*

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Dalam perkembangan dunia *Mass Rapid Transportation* (MRT) dapat dilihat kemajuan yang berarti, diantaranya adalah pembuatan kereta api cepat, *monorail* serta *tram* yang semakin canggih dan modern. Salah satu teknologinya adalah pengoperasian secara otomatis atau tanpa menggunakan masinis.

Tram sebagai salah satu tipe MRT yang populer dan telah digunakan di beberapa negara maju, merupakan salah satu alat transportasi pilihan yang akan diaplikasikan di dalam lingkungan kampus Universitas Indonesia. Diharapkan alat transportasi ini nantinya dapat menggantikan fungsi dari “bis kuning”. Menggunakan energi listrik sebagai sumber tenaganya menjadikan *tram* sebagai alat transportasi yang ramah terhadap lingkungan. Sumber tenaga listrik pada *tram* yang digunakan untuk menggerakkan motor listrik dapat dikombinasikan dengan peralatan kontrol elektronik seperti *electric drive* serta *Programmable Logic Controller* (*switching*).

Penggunaan *Programmable Logic Controller* (PLC) dibandingkan peralatan seperti *microcontroller* memiliki keunggulan seperti, kapasitas memori yang besar sehingga mampu mengendalikan suatu sistem yang kompleks, konstruksinya yang kokoh dan tahan terhadap noise serta penggunaannya yang ditujukan untuk pengendalian output dengan tegangan kerja yang tinggi (24 Volt DC dan 220 Volt AC). Salah satu contoh aplikasi PLC pada *tram* adalah panggilan otomatis *tram* dari stasiun, dimana penumpang di stasiun A yang ingin pergi menuju stasiun C dapat memanggil *tram* yang sedang berada di stasiun C. Selain itu jika terjadi perubahan deskripsi kerja dari sistem panggilan otomatis *tram* dapat diatasi dengan mudah yaitu, cukup dengan memprogram ulang PLC sesuai dengan deskripsi yang diinginkan.

Karena banyaknya keuntungan yang didapat dari penggunaan PLC, maka penggunaan peralatan ini merupakan komponen yang sangat penting dalam

pembuatan sistem transportasi menggunakan *tram* yang akan diterapkan di lingkungan kampus Universitas Indonesia Depok.

1.2 PERUMUSAN MASALAH

Dalam penyusunan pengerjaan tugas akhir ini terdapat permasalahan yang akan dihadapi, yaitu membuat program PLC yang dapat melakukan panggilan otomatis *tram* dari stasiun ataupun pemberhentian *tram* di tiap-tiap stasiun dengan komponen I/O yang sangat terbatas.

Solusi yang dikemukakan ialah dengan menempatkan komponen-komponen *input* dan *output* yang dianggap dapat mewakili kondisi sebenarnya sebagai prioritas utama, sehingga didapat gambaran yang jelas dan mendekati keadaan yang sesungguhnya dari sistem panggilan otomatis *tram*.

1.3 TUJUAN PENULISAN

Adapun tujuan pengerjaan tugas akhir ini adalah program *ladder* PLC dirancang dan digunakan untuk mengendalikan panggilan dan pemberhentian *tram* pada suatu stasiun serta diaplikasikan terhadap miniatur.

1.4 BATASAN MASALAH

Dalam tugas akhir ini terdapat beberapa hal yang menjadi batasan yaitu :

1. Program difokuskan pada panggilan dan pemberhentian *tram* pada stasiun dengan menggunakan program PLC
2. Kondisi kerja sistem merupakan kondisi kerja normal, atau dalam hal ini tidak terdapat gangguan
3. PLC yang digunakan hanya memiliki kapasitas I/O 30 poin, terdiri dari 18 *Input* dan 12 *Output*
4. Terdapat tiga buah stasiun dengan satu buah miniatur *tram* yang digunakan. Satu buah stasiun digunakan sebagai stasiun utama sebagai *home-base*.

1.5 METODOLOGI PENELITIAN

Dalam rangka mewujudkan tugas akhir ini, metodologi penelitian yang digunakan adalah sebagai berikut:

1. Metode kepustakaan, yaitu dengan membaca buku-buku referensi serta file-file digital yang berkaitan dengan tugas akhir
2. Metode eksperimen dan rancang bangun, yaitu program *ladder* PLC yang sudah dibuat diterapkan dan diaplikasikan terhadap miniatur *tram*.

1.6 SISTEMATIKA PENULISAN

Tugas akhir dibagi menjadi lima bab yang secara garis besar isinya dapat diuraikan sebagai berikut:

BAB I PENDAHULUAN

Memuat latar belakang, perumusan masalah, tujuan penulisan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II CONTROLLER PADA KERETA LISTRIK (TRAM)

Membahas tentang teori PLC, metode dan teknik yang digunakan dalam pemrograman PLC.

BAB III RANCANG BANGUN

Berisikan langkah-langkah yang harus dibuat dalam perancangan sistem, diantaranya pembuatan *lay out plant* (miniatur *tram*), pemrograman PLC seperti membuat deskripsi kerja yang diinginkan, membuat diagram alir, membuat tabel I/O yang akan digunakan, serta memprogram PLC sesuai *software* yang tersedia.

BAB IV ANALISA DAN UJICOBA PROGRAM

Membahas hasil program PLC yang telah dibuat dan dianalisa hasil program tersebut serta diujicobakan menggunakan miniatur *tram*.

BAB V KESIMPULAN

Berisikan kesimpulan dari keseluruhan Tugas Akhir.

BAB II

CONTROLLER PADA KERETA LISTRIK (TRAM)

Kereta api ringan atau disebut juga *Light Rail Transit* (LRT) adalah salah satu jenis *Mass Rapid Transportation* (MRT) yaitu, suatu sistem kereta api penumpang yang beroperasi di kawasan perkotaan, konstruksinya ringan dan dapat berjalan bersama lalu lintas lain atau dalam lintasan khusus disebut *tram*. Kereta api ringan banyak digunakan diberbagai negara di Eropa dan telah mengalami modernisasi, antara lain pengoperasiannya yang otomatis sehingga dapat dijalankan tanpa masinis, dapat berjalan pada lintasan khusus, penggunaan lantai yang rendah (sekitar 30 cm) yang disebut sebagai *Low floor LRT* untuk mempermudah naik turun penumpang. Bentuk fisik *tram* dapat dilihat pada **Gambar 2.1**, dimana sumber tenaganya menggunakan energi listrik serta dioperasikan tanpa masinis (otomatis).



Sumber : www.wikipedia.com

Gambar 2.1. Contoh bentuk fisik *tram*

Tram sebagai salah satu transportasi yang populer dan telah digunakan di beberapa negara maju, merupakan salah satu alat transportasi pilihan yang akan digunakan di dalam lingkungan kampus Universitas Indonesia. Diharapkan alat transportasi ini nantinya dapat menggantikan fungsi dari “bis kuning”. *Tram* menggunakan sumber energi listrik serta motor listrik untuk menggerakannya

sehingga dapat dikombinasikan dengan peralatan kontrol elektronik seperti *electric drive* serta *Programmable Logic Controller (switching)*.

2.1 PROGRAMMABLE LOGIC CONTROLLER (PLC)^[1]

Programmable Logic Controller (PLC) merupakan perangkat kontrol elektronik (digital) yang dirancang salah satunya untuk keperluan pengontrolan mesin-mesin listrik. Semula PLC digunakan untuk menggantikan fungsi *relay* yang banyak digunakan pada saat itu. Spesifikasi yang banyak digunakan dan dibutuhkan antara lain :

- 1) *Rugged* (kasar dan keras) tahan terhadap *noise*
- 2) Disusun secara modular sehingga memudahkan untuk menambah/ dikurangi (untuk pengembangan dan perawatan)
- 3) Mempunyai sambungan dan level sinyal yang standar
- 4) Mudah diprogram dan diprogram ulang tanpa harus menambah perangkat kontrol.

PLC menggunakan memori yang dapat diisi oleh program yang dapat menerapkan fungsi-fungsi khusus dalam elektronika digital seperti logika, sekuensial/ urutan, pewaktuan (*timer*), *counter* dan lainnya, guna mengendalikan suatu proses analog/ digital dari suatu proses.

PLC merupakan piranti berbasis *microprocessor* dan dapat dianggap sebagai komputer yang dirancang untuk tujuan pengendalian tertentu. *Console* hanya diperlukan untuk memasukan program secara *online*, yang langsung diketikkan melalui terminal pemrograman genggam (*keypad*). Satu unit *console* dapat menangani hingga beberapa unit PLC. PLC memberikan respon terhadap berbagai sensor yang dihubungkan ke *input*-nya, memutuskan proses apa yang dikerjakan berdasarkan instruksi yang telah diberikan dan diprogram ke dalam memorinya, dan memberikan ketetapan terhadap *output* yang diinginkan.

Aplikasi dari PLC sangatlah banyak untuk disebutkan, sebagai gambaran jenis peralatan yang sering digunakan adalah motor listrik, pompa, katup, mesin pengemasan, mesin perakit dan lainnya.

Untuk aplikasi-aplikasi tersebut, terdapat beberapa keuntungan yang dapat diperoleh dengan menggunakan PLC daripada menggunakan sistem kontrol

konvensional. Pertama, mereka diprogram dengan bahasa yang sederhana, yang disebut logika tangga (*ladder logic*). Bahasa ini mengambil bentuk sebagai instruksi yang mudah dipelajari, yang dimasukkan secara langsung dari *tuts* berlabel pada *console*. Kedua, masukan dan keluaran PLC dapat secara langsung dihubungkan dengan berbagai piranti seperti saklar, *relay*, dan sensor, sedangkan untuk keluaran komputer diperlukan peralatan *interface* tambahan yaitu port masukan/ keluaran (*I/O Port*). Ketiga, PLC dapat digabungkan langsung dengan sebuah PROM (*Programmable Read Only Memory*), yang tetap akan menyimpan program meskipun daya listrik telah dimatikan. Sebaliknya, untuk menggabungkan sebuah PROM dengan komputer standar, perlu dilakukan beberapa modifikasi.

Selain keuntungan yang dijelaskan diatas, PLC juga dirancang dan dikonstruksi dengan kemampuan untuk tetap dapat bekerja pada lingkungan-lingkungan yang cukup berat dan kasar, dengan kata lain suatu lingkungan yang mempunyai kondisi temperatur yang cukup tinggi, kelembaban udara yang tinggi, pengaruh vibrasi serta kondisi *noise* dan kejutan-kejutan yang timbul karena kondisi kerja mesin-mesin atau peralatan listrik lainnya. Sehingga dengan memanfaatkan PLC sebagai alat untuk melaksanakan proses kontrol mesin-mesin listrik dapat diperoleh kelebihan/ keuntungan sebagai berikut :

- 1) Dapat bekerja dengan cukup aman, handal serta fleksibel. Yaitu dapat dihubungkan/ berkomunikasi dengan peralatan kontrol lainnya
- 2) Mudah untuk mengubah program atau rancangan dari rangkaian proses kontrol
- 3) Mengurangi/ menghemat pemakaian kawat/ kabel instalasi sistem kontrol
- 4) Tidak memerlukan tempat yang luas dalam penginstalasiannya walaupun kontrol yang ditangani cukup rumit.
- 5) Relatif mudah dalam pelaksanaan perawatan dan perbaikan.
- 6) Mudah dalam melaksanakan pengembangan dan perluasan sistem kontrol proses.

Untuk menetapkan penggolongan PLC terlebih dahulu dilihat kriterianya, salah satu kriteria yang penting adalah jumlah maksimum dari port masukan/ keluaran. Kriteria ini akan memberikan informasi mengenai kemampuan PLC,

karena makin banyak I/O *port* yang dapat dikendalikan maka jumlah memori yang diperlukan juga makin besar, sehingga diperlukan pula CPU yang lebih cepat. Dengan menggunakan kriteria ini maka akan diperoleh penggolongan PLC sebagai berikut :

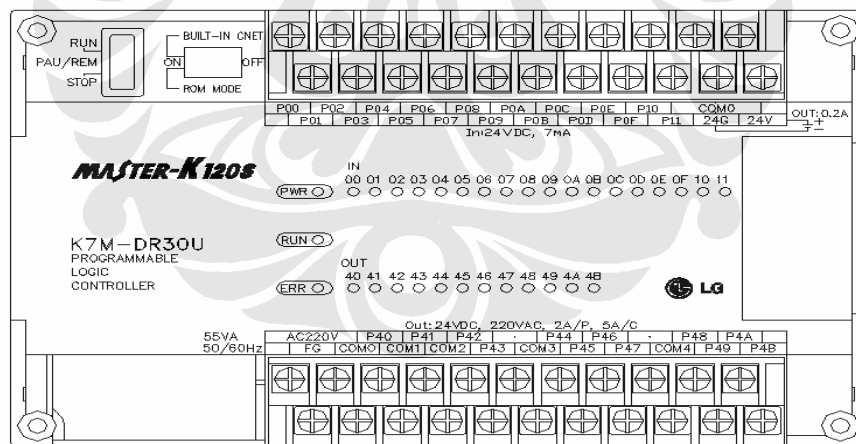
- 1) Skala mikro (15 hingga 64 *port* I/O)
- 2) Skala kecil (64 hingga 128 *port* I/O)
- 3) Skala menengah (128 hingga 512 *port* I/O)
- 4) Skala besar (lebih dari 512 *port* I/O)

PLC terdiri dari unit kontrol (CPU), memori program, rangkaian *input*, rangkaian *output* dan unit *power supply*.

2.1.1 Bagian bagian PLC

PLC memiliki tiga bagian utama, yaitu pusat pengolahan data, modul *input* dan modul *output*, selain itu juga PLC memiliki EPROM yang akan menyimpan data data/ hasil pemrograman.

Pada **Gambar 2.2** dapat dilihat bentuk fisik dari PLC dari “LG Master-K 120S 30 I/O” adalah:



Sumber : www.lgis.com

Gambar 2.2. Contoh bentuk fisik PLC “LG Master-K 120S 30 I/O”

a) Pusat Pengolahan Data (CPU)

CPU merupakan otak dari PLC, program dipanggil dari unit memori dan diproses CPU. Pemrosesan dapat disebut sebagai menjalankan program. Apa yang sebenarnya terjadi adalah bahwa program tersebut di-scan, ini berarti

bahwa program diperiksa dari awal hingga akhir dan diinformasikan baru dimasukkan. Ini sering disebut “waktu *scan*” PLC, walaupun sebenarnya lebih berkaitan dengan waktu pengoperasian program. *Scan* dari program umumnya memakan waktu beberapa milidetik, tetapi hal ini tergantung pada panjang program dan kerumitannya. Setelah *scan* yang satu selesai, *scan* berikutnya akan dimulai dengan segera. PLC membutuhkan tempat untuk menyimpan program kerja, fasilitas ini disebut unit memori atau memori saja. Peralatan memori dapat memiliki penyimpanan data jangka pendek atau jangka panjang.

b) Modul *Input* PLC

Input adalah peralatan luar yang terhubung pada modul masukan yang akan memberikan sinyal kepada PLC, bahwa sesuatu sedang terjadi diluar PLC. Masukan bisa berupa sakelar, sensor dan berbagai macam transduser. Pada PLC “LG Master-K 120S 30 I/O” masukan diberikan Kode I dengan jumlah *Input* sebanyak 18 buah P00 –P11 dan P0A-P0F, dan kontak untuk yang tersedia untuk *input* adalah NO dan NC.

c) Modul *Output* PLC

Output PLC adalah Relay atau Transistor, yang dihubungkan dengan peralatan luar. Melalui relay atau transistor ini nantinya akan menggerakkan atau mengaktifkan peralatan luar, seperti lampu, *display*, relay, kontaktor. Untuk PLC “LG Master-K 120S 30 I/O” *Output* diberikan Notasi Q dengan jumlah sebanyak 12 buah P40 – P49 dan P4A-P4B.

PLC bekerja dengan menerima data dari peralatan *input* yang merupakan saklar-saklar, tombol-tombol, sensor-sensor dan lain sebagainya. Perubahan yang terjadi pada peralatan *input* akan memberikan sinyal pada PLC, kondisi *input* tersebut akan diolah oleh PLC selanjutnya perintah-perintah dari *input* akan ditransfer oleh PLC ke *outputnya* yang kemudian dapat digunakan untuk menggerakkan mesin-mesin.

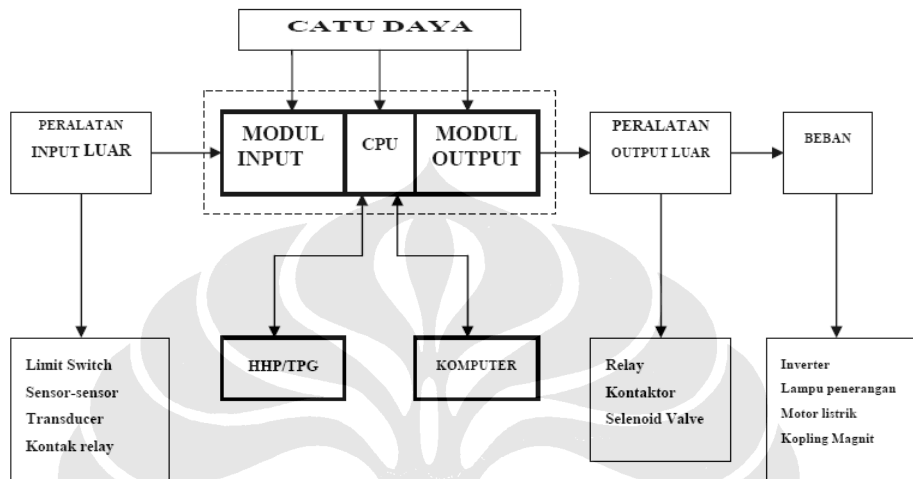
2.1.2 Prinsip Kerja

Dari definisi diatas didapat gambaran bahwa prinsip kerja PLC tetap memenuhi kriteria dari blok diagram dasar dari sistem kontrol seperti pada **Gambar 2.3**.



Gambar 2.3. Blok diagram dasar sistem kontrol PLC

Dari blok diagram dasar diatas, prinsip kerja PLC dikembangkan seperti yang terlihat pada **Gambar 2.4.**



Gambar 2.4. Blok diagram lengkap sistem kontrol PLC

2.1.3 Metode Memprogram PLC

Program adalah serangkaian perintah yang memberikan instruksi kepada PLC untuk melaksanakan tugas yang telah ditentukan. Untuk memprogram PLC dapat dilakukan dengan Cara :

1. *Instruction List*

Sistem pemrograman ini bersifat tekstual. Singkatan-singkatan khusus yang disebut *mneumonics* digunakan untuk mengidentifikasi perintah-perintah berbeda yang sedang dijalankan. Tipe ini dapat juga disebut sebagai program secara *'on line'*, unit pemrograman (*console*) ini harus dihubungkan pada PLC. Contoh: LD, OR, AND, OUT.

2. *Diagram Ladder*

Alternatif lain dari program format instruksi adalah sistem grafis yang disebut dengan *ladder*. Sistem ini menggunakan gambar grafis atau ikon-ikon untuk mewakili perintah-perintah. Pemrograman ladder pada umumnya digunakan pada perlengkapan *portable* (komputer) yang digunakan untuk

pemrograman. Perlengkapan ini memungkinkan pemrograman *offline*. Pemrograman *offline* berarti bahwa perlengkapan pemrograman tidak harus dihubungkan dengan PLC. Setelah program ditulis, PLC yang digunakan dihubungkan pada komputer dan program ditransfer ke PLC. Yang unik ladder dari satu salah produsen PLC tidak dapat langsung dimasukkan ke produsen PLC yang lain. Jadi kita harus ladder lagi dengan program pembawanya masing-masing.

3. *Sequential Function Chart* (SFC)

Perangkat lunak pemrograman terbaru dinamakan SFC (*Sequential Function Chart*). Perangkat lunak ini juga dirancang untuk digunakan pada komputer yang kompatibel dengan IBM. Dalam SFC, grafis yang ditampilkan mewakili alur logika suatu tugas, jadi program hanya berisi diagram alur dari respon yang diinginkan.

2.1.4 Perancangan Program PLC

Dalam pengerjaan tugas akhir ini terdapat beberapa langkah yang harus diperhatikan dalam perancangan program PLC, yaitu :

- 1) Menentukan *Input* dan *Output* PLC (telah ditentukan sebanyak 18 *Input* dan 12 *Output*)
- 2) Menentukan sebuah deskripsi kerja secara lengkap dan sistematis
- 3) Merubah kedalam bentuk *flowchart* untuk memudahkan pemrograman
- 4) Membuat program dalam bentuk diagram tangga (*ladder diagram*)
- 5) Memasukkan program kedalam CPU dan menyimpannya
- 6) Menguji program yang telah dibuat
- 7) Memperbaiki program yang telah dibuat
- 8) Menguji pada rangkaian yang sebenarnya dengan cara menghubungkan modul I/O pada peralatan yang sebenarnya, dalam hal ini miniatur *tram*
- 9) Menyimpan program yang telah sesuai kedalam memori PLC.

2.2 PEMROGRAMAN PLC

Untuk membentuk satu rangkaian secara logis diperlukan bermacam-macam instruksi pada PLC. Sebelum memasukan program yang telah telah dibuat PLC harus dalam kondisi *Stop*, jika PLC dalam kondisi *Run* program tidak akan bisa

masuk ke dalam memori PLC. Jika pada memori PLC masih terdapat program yang tersimpan, program tersebut akan dihapus dengan sendirinya ketika dimasukan program yang baru, atau bisa juga menghapusnya secara manual sebelum memasukan program yang baru.

Pemrograman PLC Master-K menggunakan *software* KGLWIN didalamnya terdapat suatu metoda khusus yang sederhana disebut *ladder diagram*. Dasar dari pemrograman *ladder* adalah pemahaman mengenai fungsi *relay*, kontak-kontak seperti *normaly open* (NO) dan *normaly close* (NC).

2.2.1 Instruksi PLC Master-K 120S ^[2]

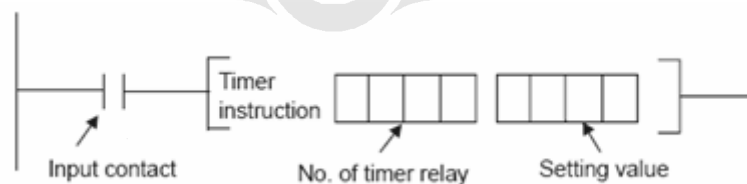
Master-K 120S memiliki instruksi program yang terdapat pada *software* KGLWIN seperti *auxiliary relay*, fungsi waktu, *counter* dan sebagainya. Berikut ini adalah beberapa instruksi yang digunakan pada pemrograman sistem panggilan otomatis *tram*.

a) *Input*

PLC akan menerima sinyal *input* baik digital (ON/OFF) maupun analog (berdasarkan nilai mV). *Input* pada tugas akhir ini sebanyak 18 *point* dan dinotasikan dengan P00-P11, P0A-P0F.

b) *Timer Relay*

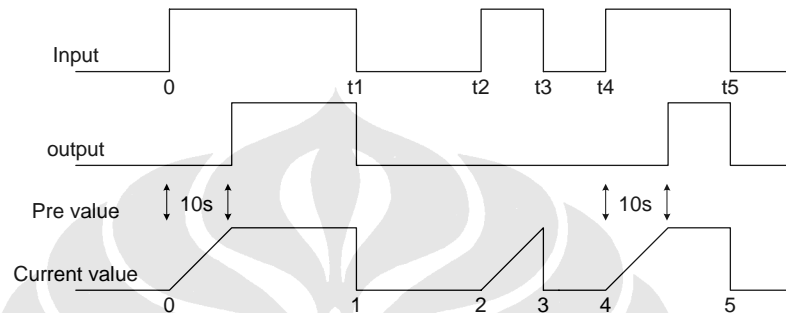
Tipe PLC Master-K memiliki *timer* dengan spesifikasi 100msec, 10msec dan 1msec. Penomoran *timer* seperti yang ditunjukkan pada **Gambar 2.5** berdasarkan spesifikasi yang dimilikinya adalah T00-T191 (100msec), T192-T250 (10msec) dan T251-255 (1msec).



Gambar 2.5. Bagian-bagian program untuk *timer*

Metode *timer* memiliki beberapa variasi sesuai dengan instruksi *timer* yang tersedia (TON, TOFF, TMR, TMON, TRTG). Maksimum *setting value*

pada *timer* adalah hFFFF untuk heksadesimal dan 65535 untuk desimal. **Gambar 2.6** adalah contoh penggunaan *timer* TON.



Gambar 2.6. *Timer* TON dengan *setting value* 10 detik

c) *Auxiliary relay/ Internal relay* (M)

Auxiliary relay adalah salah satu fasilitas yang tersedia pada PLC, dimana *internal relay* ini sangat membantu dalam pemrograman. *Internal relay* pada PLC Master-K dinotasikan dengan M dan berjumlah M00-M191.

d) *Output*

PLC akan memberikan sinyal *output* baik digital (ON/OFF) maupun analog (berdasarkan nilai mV). *Output* pada tugas akhir ini sebanyak 12 *point* dan dinotasikan dengan P40-P49, P4A-P4B.

2.2.2 Metode Y-Map (*self-holding relay*) ^[3]

Metode pemrograman Y-Map adalah salah satu teori dalam perancangan program *ladder* dalam PLC, dimana metode ini ditujukan untuk membantu *mapping* pemrograman bahasa *ladder*. Pada dasarnya teori ini ditujukan untuk merancang deskripsi dari suatu program yang memiliki cara kerja yang berurutan atau *sequential* serta memiliki siklus. Salah satu jenis metode yang sering digunakan adalah metode *self-holding relay* (SHR). *Self-holding relay* sendiri adalah suatu teknik dalam pemrograman PLC yang memiliki arti penguncian

terhadap diri koil *relay*, maksudnya kontak yang digunakan pada koil *relay* dimanfaatkan untuk mengunci koil ketika koil mendapatkan sinyal tanpa terpengaruh lamanya waktu pemberian sinyal. Metode SHR seperti yang ditunjukkan pada **Gambar 2.7** memungkinkan mengontrol berapapun jumlah *output* (beban) hanya dengan menggunakan satu buah saklar (*input*). Berikut ini adalah contoh dasar pemrograman ladder dengan metode *self-holding relay*, dimana jumlah *output* yang dikontrol oleh sebuah saklar sebanyak 5 buah. Simbol '1' digunakan untuk menyatakan kondisi ON dan simbol '0' untuk kondisi OFF.



Gambar 2.7. *Self-holding relay* pada koil M0000

Metode Y-map dapat dijadikan dasar pemrogram *ladder* dan dapat dikembangkan pada aplikasi dengan deskripsi kerja berurutan serta memiliki siklus pengulangan (*looping*).

Tabel 2.1 menjelaskan deskripsi kerja dari Y-map, bahwa kondisi ON pada *output* berurutan dengan dikontrol oleh satu *input* yaitu M. *Input* hanya memiliki kondisi ON dan OFF yang dinyatakan pada kolom status.

Tabel 2.1. Deskripsi kerja Y-map *self-holding relay* dengan 5 *output*

Input (M)		Output (Y)				
Kondisi	status	Y1	Y2	Y3	Y4	Y5
1	1	1				
2	0	1				
3	1	1	1			
4	0	1	1			
5	1	1	1	1		
6	0	1	1	1		
7	1	1	1	1	1	
8	0	1	1	1	1	
9	1	1	1	1	1	1
10	0	1	1	1	1	1
11	1	0	0	0	0	0
12	0	0	0	0	0	0

Pada **Tabel 2.2** digambarkan deskripsi kerja beberapa jenis *output* dengan menggunakan metode SHR. Deskripsi tersebut dapat diwujudkan dengan memprogram *ladder* sesuai metode SHR, yaitu program dibagi menjadi dua bagian, dimana bagian pertama merupakan *main program* sedangkan bagian yang kedua merupakan program untuk *Output*.

Tabel 2.2. Tabel macam-macam tipe *output* dari metode Y-map SHR

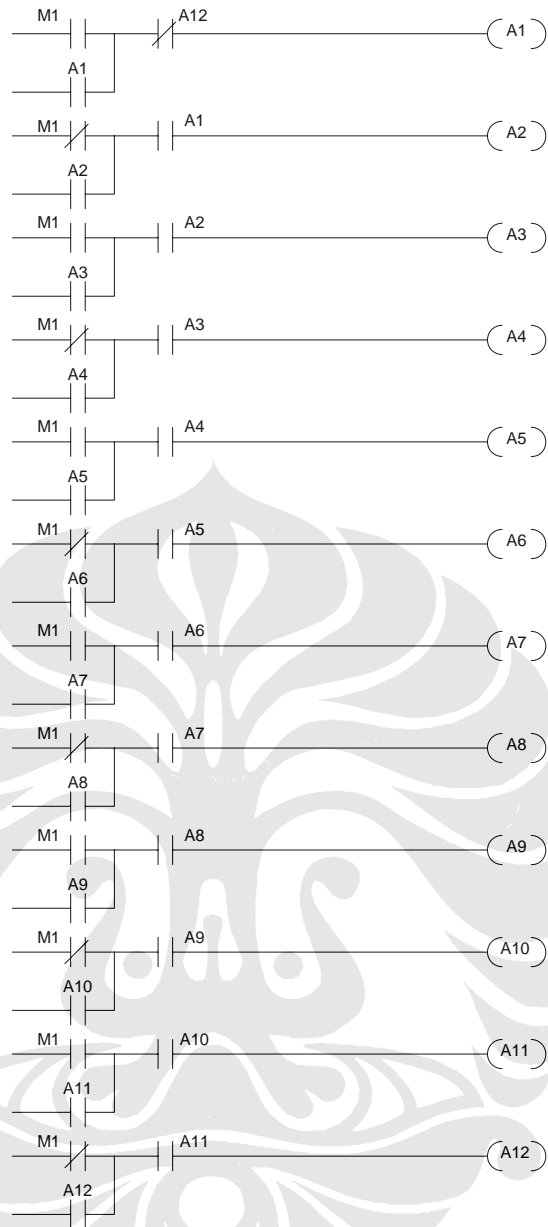
Output (Y)				
Y1	Y2	Y3	Y4	Y5
1				
1				
1	1			
1	1	1		
1	1	1	1	
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	0	0	0	0
0	0	0	0	0

Output (Y)				
Y1	Y2	Y3	Y4	Y5
1				
1				
0	1			
0	1			
	0	1		
	0	1		
		0	1	
		0	1	
			0	1
			0	1
				0
				0

Output (Y)				
Y1	Y2	Y3	Y4	Y5
		1		
		1		
	1		1	
	1		1	
1				1
1				1
	1		1	
	1		1	
		1		
		1		
		0		
		0		

Output (Y)				
Y1	Y2	Y3	Y4	Y5
1				1
1				1
	1		1	
	1		1	
		1		
		1		
	1		1	
	1		1	
1				1
1				1
0				0
0				0

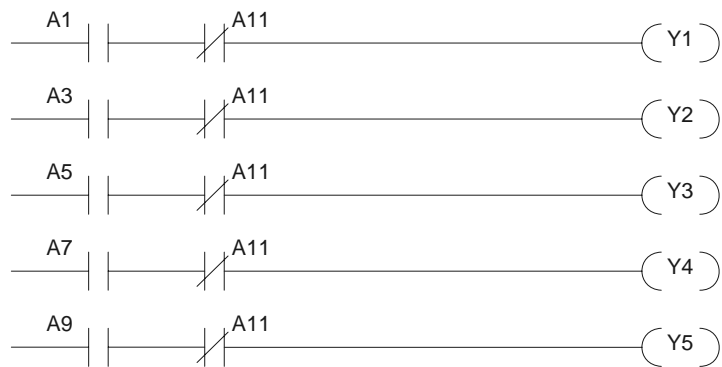
Pembagian program menjadi dua bagian tersebut dimaksudkan untuk mempermudah *user* dalam mengevaluasi program apabila terdapat kesalahan didalamnya. Program *ladder* lengkap untuk metode SHR dengan 5 *output* diberikan pada **Gambar 2.8** dibawah.



Gambar 2.8. Program utama kontrol proses metode Y-map SHR

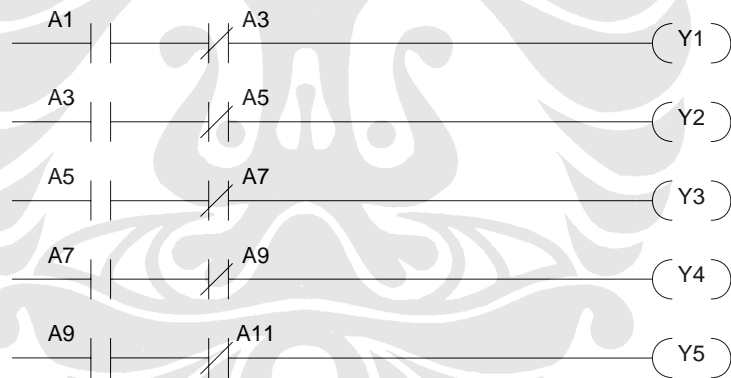
Program pada **Gambar 2.8** merupakan program utama dari keseluruhan program dengan metode Y-map SHR. Program ini berisikan kontrol yang akan mempengaruhi kondisi *output*.

Program pada **Gambar 2.9** merupakan program untuk *output* tipe (a), dimana cara kerja *output* adalah ON secara berurutan atau satu per satu lalu kemudian akan OFF secara serentak.



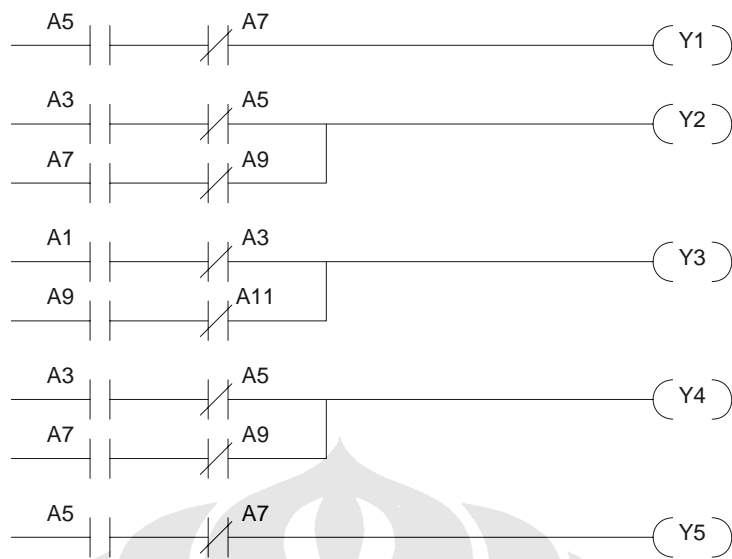
Gambar 2.9. *Output tipe (a)*

Program pada **Gambar 2.10** merupakan program untuk *output* tipe (b), dimana cara kerja *output* hampir sama dengan *output* tipe (a) ON secara berurutan, yang membedakan adalah ketika *output* selanjutnya yang ON maka *output* sebelumnya akan OFF.



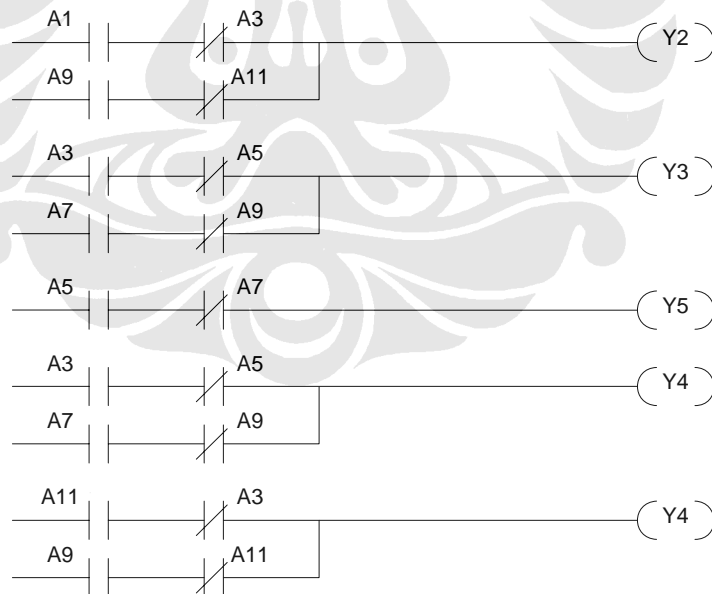
Gambar 2.10. *Output tipe (b)*

Sementara untuk program pada **Gambar 2.11** merupakan program untuk *output* tipe (c), cara kerja *output* sesuai dengan tabel dimana *output* akan aktif dari terminal *input* paling kiri dan paling kanan menuju ke tengah.



Gambar 2.11. *Output tipe (c)*

Program pada **Gambar 2.12** merupakan program untuk *output* tipe (d), cara kerja *output* sesuai dengan tabel dimana *output* akan aktif dari terminal *input* paling tengah menuju terminal *input* paling kanan dan kiri.



Gambar 2.12. *Output tipe (d)*

BAB III

RANCANG BANGUN

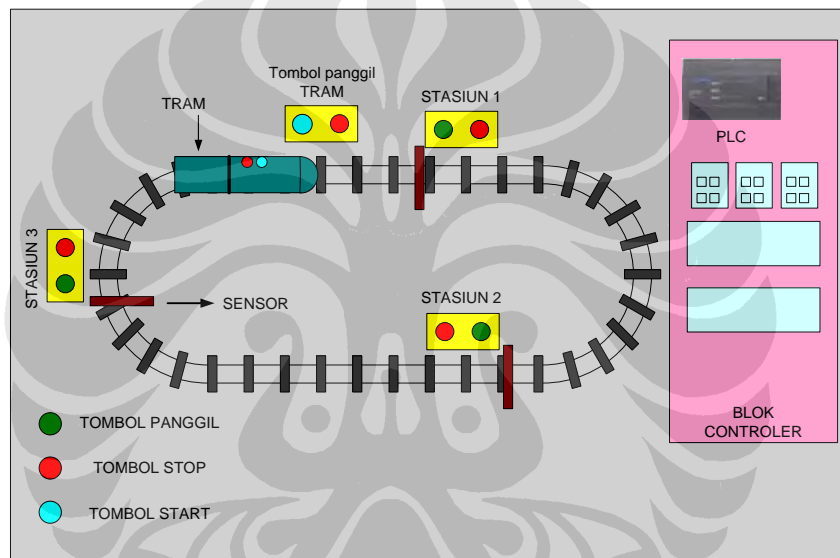
Dalam perancangan suatu program PLC, salah satu hal yang paling utama adalah mengetahui deskripsi kerja yang diinginkan serta harus diketahui bentuk dari plant itu sendiri, sehingga memudahkan perancang dalam merancang kontrolnya.

3.1 LAY-OUT

Pada rancang bangun *plant* sistem panggilan otomatis *tram* ini terdiri atas beberapa komponen yang akan digunakan, diantaranya:

1. Kereta mainan. Digunakan sebagai miniatur *tram*, dimana bagian terpenting dari miniatur kereta disini adalah motornya yang dijadikan beban/ *output* untuk dikontrol oleh PLC. Memiliki tegangan kerja sebesar 3 Volt DC dengan arus yang mengalir sekitar 20mA.
2. Rel yang berfungsi sebagai jalur lintasan yang digunakan *tram* sebagai “jalan raya-nya”. Pada *plant* miniatur dipasang beberapa kawat dibawah rel sebagai media penghantar listrik dengan tegangan 3 Volt DC.
3. Sensor sebagai *switch* yang dimanfaatkan sebagai *Input* PLC. Sensor disini digunakan untuk signal kontrol pemberhentian *tram*. Menggunakan sensor cahaya LED.
4. *Push-button* tanpa penguncian yang digunakan untuk memanggil, menjalankan, serta menghentikan *tram*.
5. *Power supply*. Komponen ini digunakan sebagai sumber daya pada output PLC, tersedia dengan tegangan sebesar 5 Volt DC dan 24 Volt DC.
6. Blok *controller* dan panel. Terdiri atas komponen untuk mengendalikan sistem yaitu PLC serta beberapa terminal yang digunakan untuk penginstalasian kabel (*wiring*).

Dari *lay-out plant tram* pada **Gambar 3.1** terlihat jumlah stasiun yang digunakan sebanyak tiga buah. Walaupun dalam kenyatannya Universitas Indonesia memiliki 14 buah *shelter* pemberhentian, dimana 8 buah *shelter* diantaranya terletak di depan fakultas, akan tetapi secara program tiga buah stasiun dirasakan cukup untuk mewakili jumlah *shelter* yang banyak tersebut. Hal ini dikarenakan keterbatasan jumlah I/ O PLC Master-K yang digunakan. Program yang dibuat diharapkan dapat mengendalikan panggilan dan pemberhentian *tram* secara otomatis pada suatu stasiun dengan menggunakan PLC, dimana ke depan nantinya akan diterapkan di lingkungan kampus Universitas Indonesia Depok.



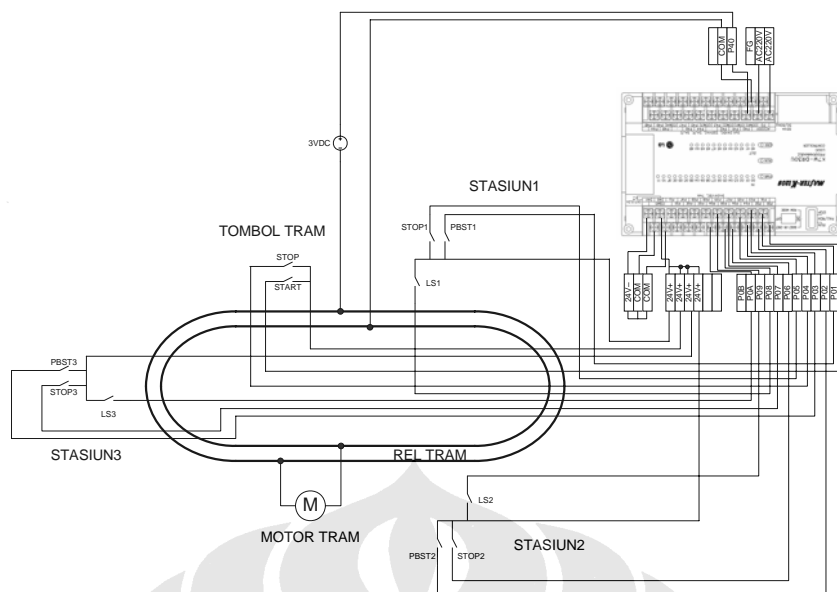
Gambar 3.1. *Lay-out plant* miniatur tram

Sebelum melakukan perancangan program pada PLC, langkah awal yang harus dilakukan adalah menentukan komponen I/O yang digunakan. Setelah itu baru dapat diketahui jumlah I/O yang diperlukan untuk melakukan pemrograman dengan PLC. Pada komponen *Input* PLC terdiri dari sensor serta *push-button* yang akan mengendalikan *Output* PLC, dalam hal ini motor yang terdapat pada *tram*. Dari perencanaan awal yang telah dibuat, didapat jumlah I/O yang digunakan dalam PLC. Dibawah ini (**Tabel 3.1**) merupakan tabel I/O PLC pada sistem panggilan otomatis *tram*.

Tabel 3.1. *Input dan Output PLC*

<i>Input PLC</i>		<i>Output PLC</i>	
Simbol	Keterangan	Simbol	Keterangan
PBST1	Tombol panggilan stasiun 1	M1	Motor <i>tram</i>
PBST2	Tombol panggilan stasiun 2		
PBST3	Tombol panggilan stasiun 3		
LS1	Sensor stasiun 1		
LS2	Sensor stasiun 2		
LS3	Sensor stasiun 3		
START	Tombol jalankan <i>tram</i>		
STOP	Tombol hentikan sementara <i>tram</i> (tombol di dalam <i>tram</i>)		
STOP1	Tombol hentikan sementara <i>tram</i> (tombol pada stasiun 1)		
STOP2	Tombol hentikan sementara <i>tram</i> (tombol pada stasiun 2)		
STOP3	Tombol hentikan sementara <i>tram</i> (tombol pada stasiun 3)		

Setelah diketahui jumlah komponen I/O yang digunakan maka langkah selanjutnya seluruh komponen baik *push-button* maupun sensor dihubungkan dengan modul masukan PLC. Sementara modul keluaran dihubungkan dengan rel yang telah dipasang kawat tembaga sebagai media penghantar arus listrik menuju motor yang terdapat pada miniatur *tram*. Langkah ini dapat ditunjukkan pada **Gambar 3.2**, yaitu *wiring diagram* dari miniatur *tram* yang akan dibuat.



Gambar 3.2. *Wiring diagram miniatur tram*

Setelah perangkat *hardware* telah terpasang seperti yang ditunjukkan pada **Gambar 3.3** maka dapat dilakukan pengerjaan dengan *software*.

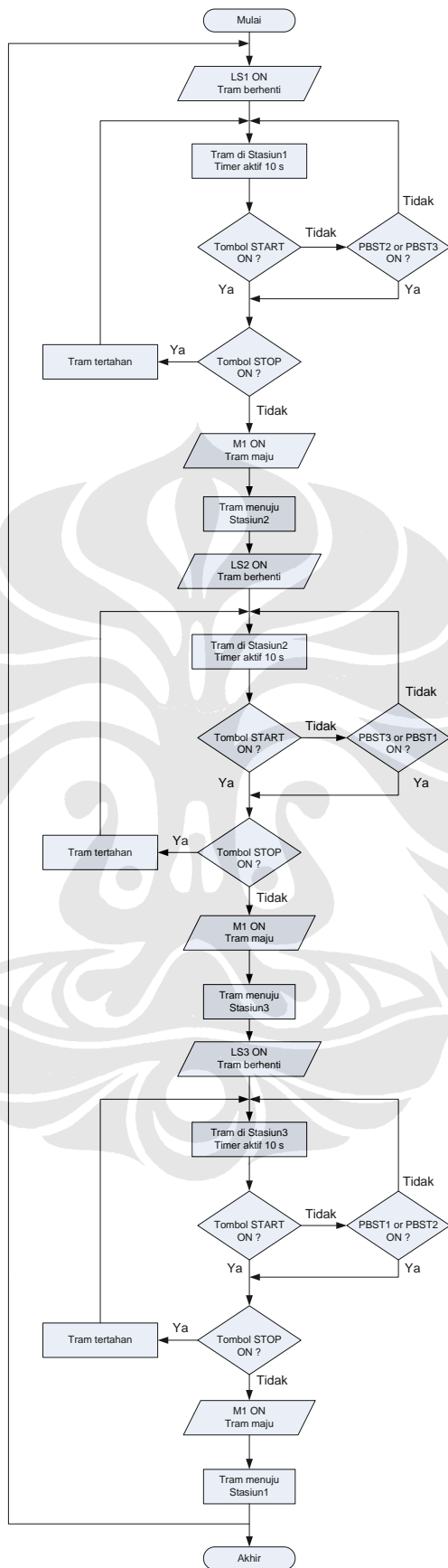


Gambar 3.3. *Miniatur tram*

3.2 DIAGRAM ALIR

Langkah berikutnya yaitu deskripsi kerja awal dari sistem diubah ke dalam diagram alir.

Pembuatan diagram alir yang ditunjukkan pada **Gambar 3.4** sangat dibutuhkan sebelum melakukan perancangan program pada PLC, karena akan diperoleh deskripsi kerja dari sistem yang lebih sederhana jika dibandingkan dengan penggunaan kata-kata untuk menjelaskan sistem secara keseluruhan.



Gambar 3.4. Diagram alir sistem panggilan otomatis tram

3.3 DESKRIPSI KERJA TRAM

Dari diagram alir yang telah dibuat maka deskripsi kerja dari sistem panggilan otomatis *tram* adalah:

1. Kondisi awal/ *default* sistem panggilan *tram*, *tram* berada di stasiun 1. Dimana stasiun 1 berfungsi sebagai *home base* dari *tram*.
2. Ketika terdapat panggilan dari salah satu stasiun 2 atau stasiun 3 (PBST2 atau PBST3 kondisi ON) ataupun tombol START kondisi ON maka *tram* akan bergerak menuju stasiun 2. Kondisi ini memiliki persyaratan dimana *tram* akan bergerak jika tombol STOP dalam kondisi OFF pada stasiun 1. *tram* hanya akan berhenti/ tertahan selama 10 detik pada stasiun 1 apabila LS1 ON dan tombol STOP ON.
3. Setelah *tram* bergerak menuju stasiun 2 lalu menyentuh LS2 yang menyebabkan kondisi LS2 menjadi ON maka *tram* akan berhenti secara otomatis.
4. Saat ini *tram* telah berada di stasiun 2 dan waktu berhenti di-set dengan nilai 10 detik. Jika masih terdapat panggilan dari stasiun 3 ataupun dari stasiun 1 (PBST3 atau PBST1 kondisi ON) maka setelah waktu 10 detik diatas, *tram* akan bergerak menuju stasiun 3. Akan tetapi jika tidak terdapat panggilan dari salah satu stasiun tersebut baik stasiun 3 atau stasiun 1, maka *tram* tetap berada di stasiun 2. *Tram* hanya akan bergerak jika tombol START ataupun salah satu dari tombol PBST3 dan PBST1 kondisinya ON.
5. Ketika terdapat panggilan dari salah satu stasiun 3 dan stasiun 1 (PBST2 atau PBST3 kondisi ON) ataupun tombol START kondisi ON maka *tram* akan bergerak menuju stasiun 3. Kondisi ini memiliki persyaratan dimana *tram* akan bergerak jika tombol STOP dalam kondisi OFF pada stasiun 2. *tram* hanya akan berhenti apabila LS1 ON dan tombol STOP ON.
6. Setelah *tram* bergerak menuju stasiun 3 lalu menyentuh LS3 yang menyebabkan kondisi LS3 menjadi ON maka *tram* akan berhenti secara otomatis.
7. Saat ini *tram* telah berada di stasiun 3 dan waktu berhenti di-set dengan nilai 10 detik. Jika masih terdapat panggilan dari stasiun 1 ataupun dari stasiun 2 (PBST1 atau PBST2 kondisi ON) maka setelah waktu 10 detik diatas, *tram*

akan bergerak menuju stasiun 1. Akan tetapi jika tidak terdapat panggilan dari salah satu stasiun tersebut baik stasiun 1 atau stasiun 2, maka *tram* tetap berada di stasiun 3. *Tram* hanya akan bergerak jika tombol START ataupun salah satu dari tombol PBST3 dan PBST1 kondisinya ON.

8. Ketika terdapat panggilan dari salah satu stasiun 1 atau stasiun 2 (PBST1 atau PBST2 kondisi ON) ataupun tombol START kondisi ON maka *tram* akan bergerak menuju stasiun 1. Kondisi ini memiliki persyaratan dimana *tram* akan bergerak jika tombol STOP dalam kondisi OFF pada stasiun 1. *tram* hanya akan berhenti apabila LS1 ON dan tombol STOP ON.
9. Setelah *tram* bergerak menuju stasiun 1 lalu menyentuh LS1 yang menyebabkan kondisi LS1 menjadi ON maka *tram* akan berhenti secara otomatis.
10. Saat ini *tram* telah berada di stasiun 1 kembali dan waktu berhenti di-set dengan nilai 10 detik. Jika masih terdapat panggilan dari stasiun 2 ataupun dari stasiun 3 (PBST2 atau PBST3 kondisi ON) maka setelah waktu 10 detik diatas, *tram* akan bergerak menuju stasiun 2. Akan tetapi jika tidak terdapat panggilan dari salah satu stasiun tersebut baik stasiun 2 atau stasiun 3, maka *tram* tetap berada di stasiun 1. *Tram* hanya akan bergerak jika tombol START ataupun salah satu dari tombol PBST2 dan PBST3 kondisinya ON.
11. Tombol STOP berfungsi untuk menahan *tram* selama masih berada di stasiun dengan waktu 10 detik. Ketika *tram* sudah tidak berada di stasiun maka tombol STOP tidak dapat digunakan.
12. Tombol START digunakan untuk menjalankan *tram* secara langsung guna menghindari waktu tunggu 10 detik selama distasiun.
13. Selanjutnya proses akan kembali ke no 2.
14. Proses kontrol *tram* ini dibuat secara berurutan, dimana setiap terdapat stasiun *tram* akan berhenti selama 10 detik serta *tram* akan tetap berada di stasiun terakhir yang ditujunya selama tidak terdapat panggilan dari stasiun lain ataupun tidak dijalankan. Selain itu pergerakan *tram* adalah satu arah dengan bentuk putaran lingkaran, yaitu dari stasiun 1 – stasiun 2 – stasiun 3 dan kembali ke stasiun 1.

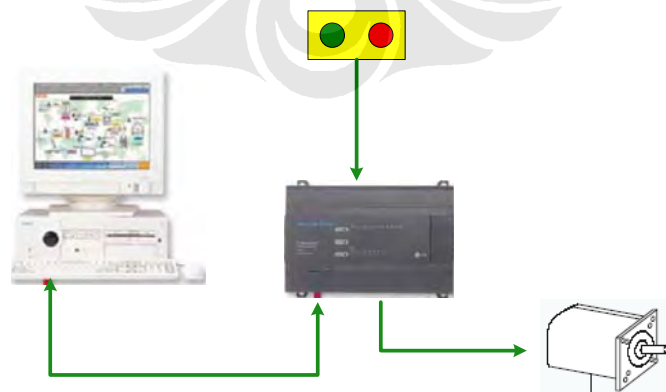
3.4 ALGORITMA PROGRAM

Dari deskripsi kerja yang telah ditentukan maka algoritma yang digunakan dalam perancangan program adalah sebagai berikut:

1. *Input-input* yang dapat mempengaruhi kondisi kerja *tram* seperti menjalankan dan memberhentikan *tram* ditentukan terlebih dahulu
2. PBST1, PBST2, PBST3, START adalah input, fungsi utamanya untuk menjalankan *tram*
3. LS1, LS2, LS3, STOP adalah input, fungsi utamanya untuk memberhentikan *tram*
4. Program dibagi menjadi beberapa sub-program sesuai dengan jumlah stasiun yang digunakan dan menggunakan metode Y-Map
5. Stasiun1 digunakan sebagai acuan, dimana program pertama yang aktif adalah program untuk kondisi di stasiun1
6. Setelah bagian program akhir (kondisi stasiun3) aktif maka digunakan program *reset* yang dapat memulihkan program secara keseluruhan untuk dapat di-*looping* kembali ke kondisi awal.

3.5 BLOK DIAGRAM

Blok diagram yang digunakan pada sistem panggilan *tram* dapat dijelaskan seperti pada **Gambar 3.5.** sesuai dengan dasar teori PLC yang dibahas pada Bab II.



Gambar 3.5. Blok Diagram Sistem Panggilan *Tram*

Perangkat komputer digunakan untuk membuat program *ladder* dengan *software* KGLWIN untuk PLC tipe Master-K 120S. Sedangkan kabel RS-232C merupakan *interface* yang digunakan untuk menghubungkan komputer dengan PLC. *Input* terdiri dari peralatan seperti tombol-tekan serta sensor-sensor yang dipasang pada stasiun dan *tram*. Peralatan input digunakan untuk mengontrol kondisi kerja daripada *output*. *Output* PLC dapat dihubungkan langsung terhadap beban dalam hal ini adalah motor sebagai penggerak *tram* dikarenakan arus yang mengalir pada beban yang melalui kontak PLC masih dibawah nilai 2A.

Selain pengoperasian melalui tombol-tekan ataupun sensor yang terdapat pada stasiun, kondisi kerja daripada *output* dapat dikontrol ataupun dioperasikan secara langsung melalui komputer. Salah satu cara yang digunakan adalah dengan memanfaatkan fungsi *monitoring* pada *software* KGLWIN serta merubah posisi kontak pada program *ladder*.

Pada **Gambar 3.6** ditunjukkan dua buah tombol yang terdapat pada stasiun yang digunakan untuk pengendalian *tram*.



Gambar 3.6. Tombol yang terdapat pada stasiun

BAB IV

ANALISA DAN UJICOBA PROGRAM

Dalam membuat program kontrol pada PLC Master-K terlebih dahulu harus menentukan bagian dari sistem kontrol yang akan diproses didalamnya, dengan cara menentukan *input* dan *output* yang akan digunakan PLC. Selain itu perlu juga diketahui instruksi-instruksi yang terdapat didalam PLC tersebut untuk memudahkan perancangan dari program kontrol ini, yaitu instruksi dasar *ladder* dan beberapa instruksi lainnya sebagai penunjang seperti yang telah dijelaskan pada Bab II.

Dalam pembuatan program dari komputer diusahakan agar program tersebut mudah dibaca sehingga apabila terjadi kesalahan mudah memperbaikinya, selain itu mudah dalam memonitor kerja program tersebut.

Dalam pengerjaan tugas akhir ini digunakan peralatan utama dan penunjang dengan spesifikasi sebagai berikut:

1. PLC Master-K 120S
Merupakan alat kontrol yang dapat diprogram, memiliki jumlah I/ O sebanyak 30 point (18 *Input* dan 12 *Output*) serta memiliki kapasitas program sebanyak 10000 langkah. Terdapat port serial yang dapat dihubungkan dengan komputer.
2. Komputer
Digunakan untuk membuka aplikasi KGLWIN dengan meng-*install* terlebih dahulu ke dalam komputer. KGLWIN adalah software yang digunakan untuk memprogram PLC dengan *operating system* minimal windows95, serta memiliki port serial (COM1 atau COM2).
3. Kabel RS232C
Kabel yang digunakan sebagai *Interface* antara komputer dengan PLC.
4. Komponen *Input*
Terdiri dari *push-button* tanpa penguncian serta sensor. Digunakan untuk memberikan sinyal kepada terminal *input* PLC dengan kondisi ON dan OFF.

5. Komponen *Output*

Beban dari dinamo motor kereta yang akan mengaktifkan miniatur *tram*.

4.1 PROGRAM *LADDER* SISTEM PANGGILAN OTOMATIS *TRAM*

Pada program *ladder* yang telah dibuat dapat diberikan penjelasan program yaitu, program terdiri dari tiga bagian. Bagian pertama merupakan program yang dikhususkan untuk *input*, tombol-tekan dan sensor diletakan pada bagian ini. Bagian kedua adalah program inti, berisikan program untuk mendapatkan deskripsi sesuai yang diinginkan. Bagian terakhir adalah program untuk mengontrol *output*. Tujuan utama dari pembagian ini adalah untuk mendapatkan program yang baik yaitu, dapat dibaca dengan mudah oleh siapapun, memudahkan dalam *trouble-shooting* dan pengembangan program selanjutnya, serta dapat membantu memonitor kerja program.

Program sistem panggilan otomatis *tram* yang telah dibuat total terdiri dari 135 langkah dan memakan memori sekitar 132kB. Program lengkap *ladder* untuk kontrol panggilan otomatis *tram* dapat dilihat pada halaman lampiran.

4.2 ANALISA PROGRAM

Pada bagian ini akan dijelaskan program *ladder* yang dibuat dan dianalisa langkah demi langkah untuk mendapatkan deskripsi kerja yang diinginkan yang telah dibahas pada Bab III.

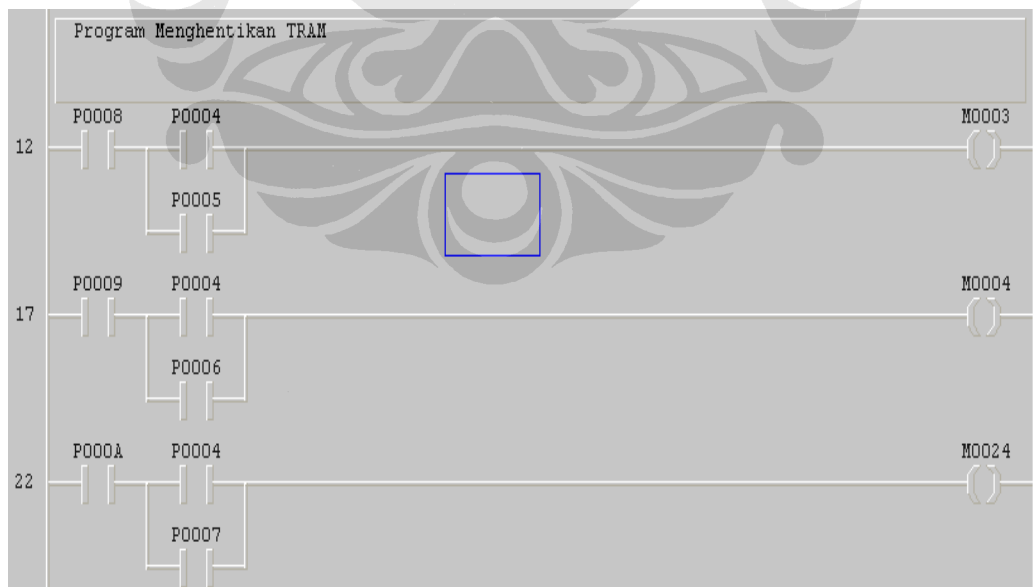
4.2.1 Program Kontrol *Input*

Pada program **Gambar 4.1** dapat dijelaskan, PBST adalah tombol panggilan yang terdapat pada tiap-tiap stasiun. Ketika tombol tersebut ditekan maka akan mengakibatkan aktifnya koil pada *internal relay* (R) dan akan mengaktifkan kontakannya. Kontak tersebut dimanfaatkan untuk melakukan penguncian diri (*self-holding relay*) SHR. LS merupakan sensor yang berada di stasiun berupa kontak *Normally Close* (NC) yang dimanfaatkan untuk melepaskan penguncian ketika sensor aktif akibat adanya kereta pada stasiun yang bersangkutan.



Gambar 4.1. Program kontrol panggilan *Push-button*

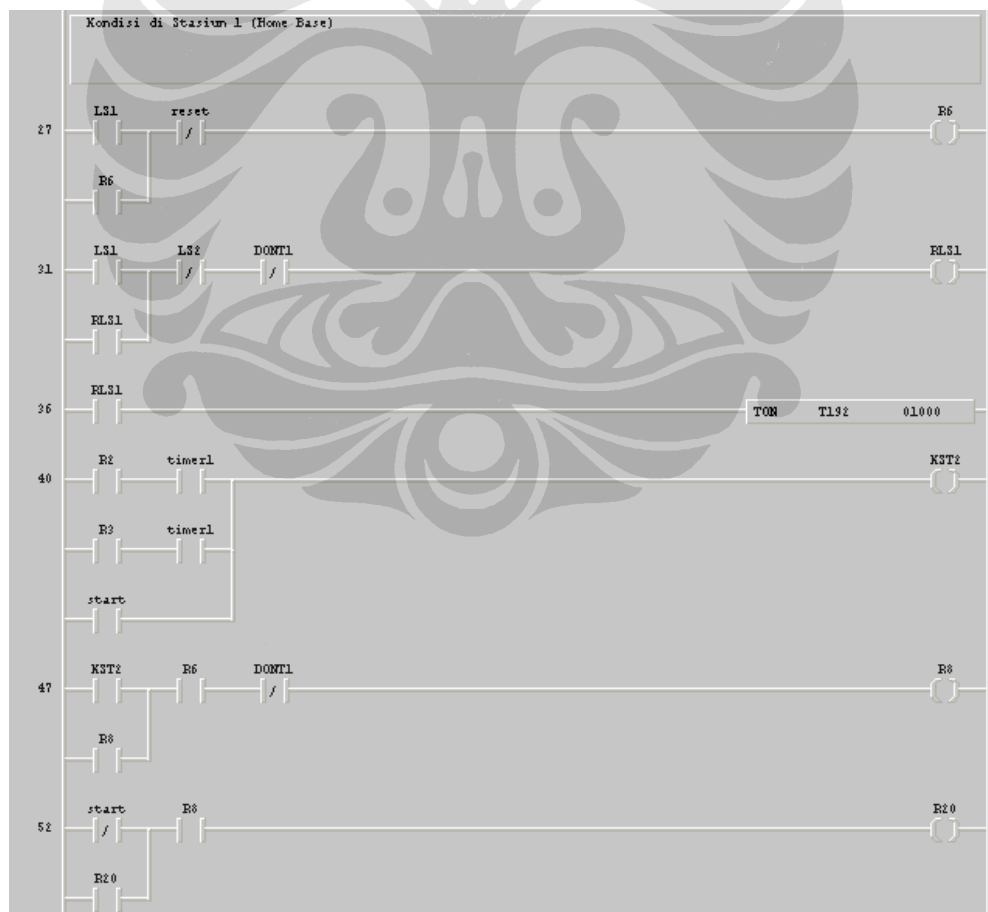
Program pada **Gambar 4.2** merupakan program untuk menghentikan *tram* yang berfungsi untuk membuat *tram* berhenti pada stasiun, *input* tersebut berupa sensor dan tombol STOP. Koil DONT1 berfungsi untuk menghentikan *tram* dari tombol STOP yang terdapat di dalam *tram*. Sedangkan DONT2 berfungsi untuk menghentikan *tram* dari tombol STOP yang berada di stasiun.



Gambar 4.2. Program kontrol menghentikan *tram*

4.2.2 Program Kontrol Utama

Program pada **Gambar 4.3** adalah semua kondisi yang terjadi di stasiun1. pada langkah ke-27, koil dan kontak R6 digunakan sebagai SHR setelah sensor LS1 aktif. Selain untuk menghentikan *tram*, R6 pula yang menjadi syarat utama dapat aktifnya R8 pada langkah ke 47, dimana kondisi ini disebut juga kondisi berurutan. Fungsi kontak NC *reset* berguna untuk melepaskan penguncian pada tiap-tiap langkah yang terdapat metode SHR-nya. Pada langkah-31 RLS1 digunakan untuk mengaktifkan *timer1* TON yang di-set waktunya selama 10 detik, sementara DONT1 dan DONT2 dari tombol STOP berfungsi untuk me-*reset* ataupun melepaskan penguncian secara sementara pada RLS1 sehingga menyebabkan *timer* menghitung kembali. Sementara fungsi LS2 adalah melepaskan penguncian RLS1 secara permanen dan me-*reset* *timer1* karena sekarang *tram* sudah berada di stasiun2 yang dapat dilihat dari aktifnya sensor LS2.



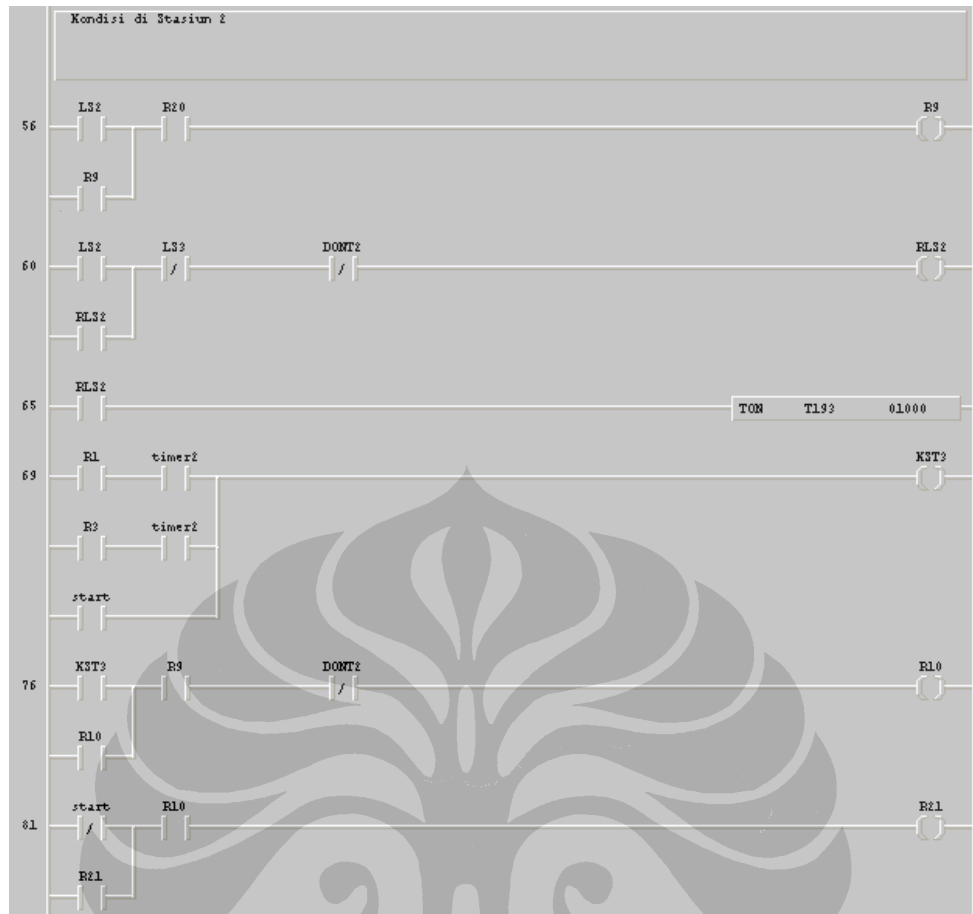
Gambar 4.3. Program kontrol *tram* di stasiun1

Pada langkah ke-40 koil KST2 digunakan sebagai syarat mengaktifkan R8 yang dapat membuat kondisi *output* motor ON. Sementara kontak R2 dan R3 yang dirangkai seri dengan kontak *timer1* dan diparalel dengan kontak START berfungsi untuk mengaktifkan KST2, apabila salah satu dari ketiga cabang rangkaian yang diparalel aktif.

Sementara pada langkah ke-47 koil R8 digunakan untuk mengaktifkan *output* motor. Kondisi ini ditentukan oleh kontak R6 dan KST2, sementara kontak DONT1 dan DONT2 digunakan sebagai *input* untuk melepaskan penguncian pada koil R8 yang diakibatkan aktifnya tombol STOP. Kontak NC pada langkah ke-52 merupakan *input* dari tombol START yang berfungsi untuk mengaktifkan koil R20, dimana kontak koil tersebut digunakan untuk mengaktifkan koil R9 yang bekerja secara berurutan.

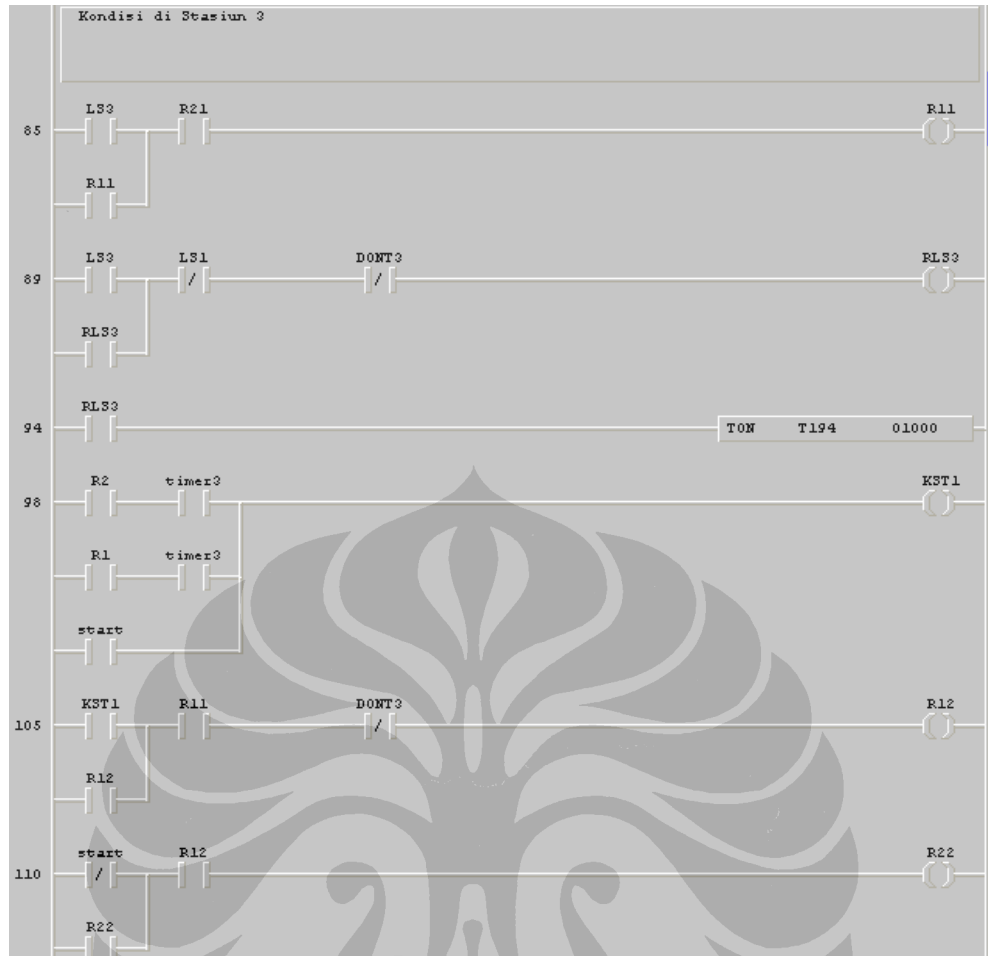
Dari penjelasan program yang mengatur kondisi pada stasiun1, terlihat jelas bahwa cara kerja program ini banyak menggunakan metode *self-holding relay* serta kerja berurutan untuk mengaktifkan koil pada langkah selanjutnya. Yang perlu diperhatikan adalah kondisi dimana *push-button* ditekan, seberapa lama penekanan dilakukan yang akan mengakibatkan efek pengiriman sinyal *input* ke dalam PLC. Untuk menghindari pemahaman yang salah dalam pemrograman PLC, maka logika yang harus digunakan adalah harus diperhatikan bahwa *input* memiliki dua kondisi yaitu ON dan OFF dengan tanpa mengabaikan seberapa lama kondisi ON dan OFF tersebut aktif.

Pada dasarnya program selanjutnya (**Gambar 4.4**) sama seperti program kontrol stasiun1, yang membedakan adalah penomoran dan pemberian variabel pada *input* dan termasuk didalamnya mengontrol fungsi *internal relay* yang berbeda. Masih sama dengan fungsi *timer* TON seperti program kontrol di stasiun1, pada kontrol stasiun2 TON yang digunakan memiliki penomoran T193 yang berarti *timer* disini memiliki kelipatan pengali 10ms sehingga jika dikalikan dengan nilai yang diberi maka T193 memiliki waktu 10 detik.



Gambar 4.4. Program kontrol *tram* di stasiun2

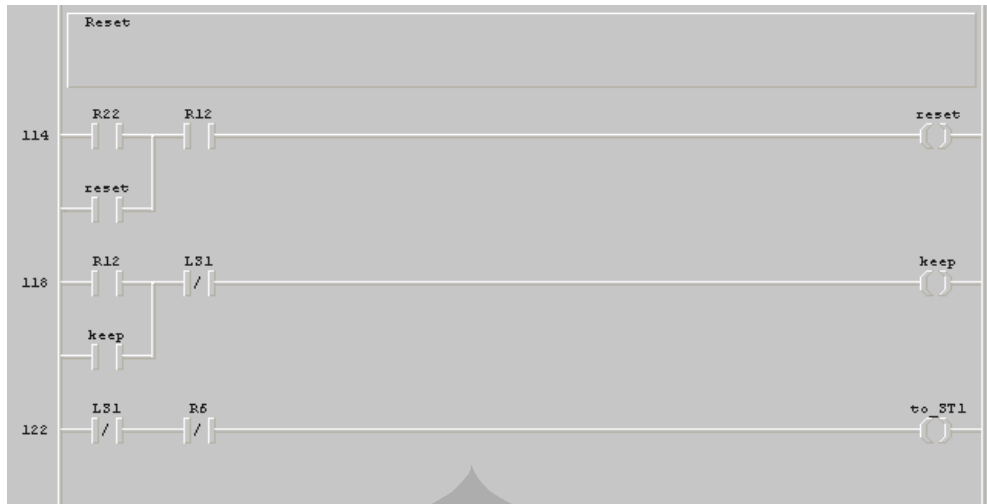
Program dibawah (**Gambar 4.5**) sama seperti program kontrol stasiun1 dan stasiun2, yang membedakan adalah penomoran dan pemberian variabel pada *input* dan termasuk didalamnya mengontrol fungsi *internal relay* yang berbeda. Masih sama dengan fungsi *timer* TON seperti program kontrol di stasiun1 dan stasiun2, pada kontrol stasiun3 TON yang digunakan memiliki penomoran T194 yang berarti *timer* disini memiliki kelipatan pengali 10ms sehingga jika dikalikan dengan nilai yang diberi maka T194 memiliki waktu 10 detik



Gambar 4.5. Program kontrol *tram* di stasiun3

Program *me-reset* seperti yang terlihat pada **Gambar 4.6** dibawah merupakan salah satu program yang penting untuk mengembalikan atau menormalisir program seperti *default* awal. Fungsi yang paling penting adalah aktifnya koil reset yang akan melepaskan penguncian SHR pada R6. Lepasnya penguncian pada R6 akan menyebabkan lepasnya penguncian beberapa relay dibawahnya, inilah yang dimaksud dengan menormalisir program.

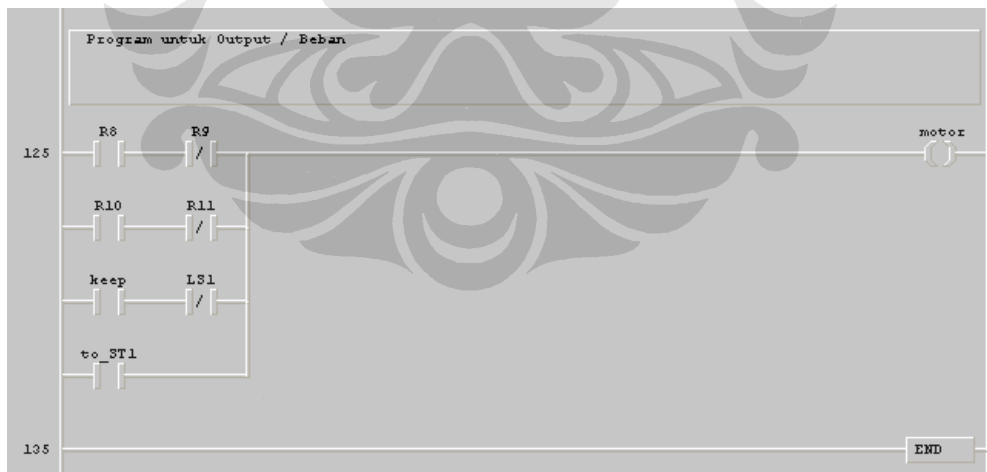
Pada langkah ke-122 merupakan bagian program yang diperuntukan untuk mengatasi apabila terjadi kehilangan daya pada PLC yang dapat mengakibatkan program kembali ke default awal. Apabila hal ini terjadi maka program ini akan mengaktifkan *output* dan menuju stasiun1 sebagai *home-base*, jadi *tram* akan meluncur menuju stasiun1.



Gambar 4.6. Program me-reset atau melepaskan penguncian SHR

4.2.3 Program Kontrol Output

Pada intinya keseluruhan program yang sudah dijelaskan sebelumnya digunakan untuk mengontrol bagian program ini (**Gambar 4.7**). Pada langkah ke-125 R8 digunakan untuk menjalankan *tram*, hal tersebut menandakan *tram* berada di stasiun1 dan siap meluncur menuju stasiun2 ketika kontak NO pada R8 aktif. Sementara apabila kontak NC pada R9 aktif menandakan kondisi dimana *tram* sudah berada di stasiun2.



Gambar 4.7. Program kontrol output

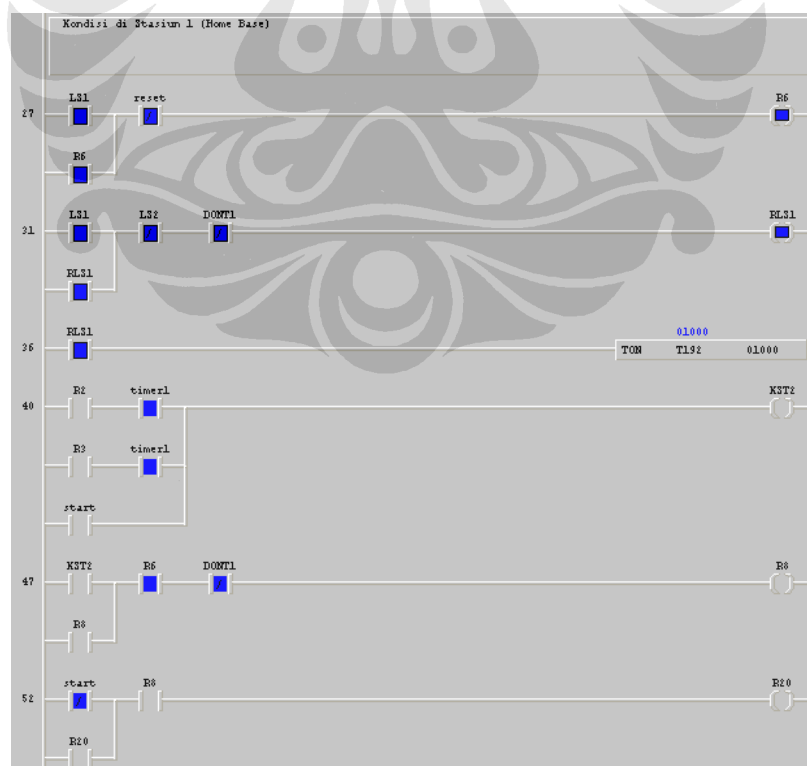
R10 digunakan untuk menjalankan *tram* dari stasiun2 menuju stasiun3, hal tersebut menandakan *tram* berada di stasiun2 dan siap meluncur menuju stasiun2

ketika kontak NO pada R10 aktif. Sementara apabila kontak NC pada R11 aktif menandakan kondisi dimana *tram* sudah berada di stasiun3.

KEEP digunakan untuk menjalankan *tram* dari stasiun3 untuk kembali menuju stasiun1, hal tersebut menandakan *tram* berada di stasiun3 dan siap meluncur menuju stasiun1 ketika kontak NO pada KEEP aktif. Sementara apabila kontak NC pada LS1 aktif menandakan kondisi dimana *tram* sudah berada di stasiun1. Sementara kontak to_ST1 digunakan untuk menanggulangi kondisi apabila PLC kehilangan daya.

4.3 UJICOBA PROGRAM

Pada bagian ini akan dijelaskan program yang dibuat dan diujicobakan secara langsung terhadap miniatur *tram*. Ujicoba dilakukan dengan berbagai kondisi/ skenario yang mungkin terjadi pada kondisi yang sebenarnya. Setelah PLC diaktifkan (posisi *run*), kondisi awal yang terjadi adalah *tram* bergerak menuju stasiun1 (apabila *tram* tidak berada di stasiun1). Pada **Gambar 4.8** ditunjukkan kondisi awal program pada stasiun1, dengan melihat kondisi LS1 dapat diketahui lokasi *tram* berada yaitu di stasiun1.



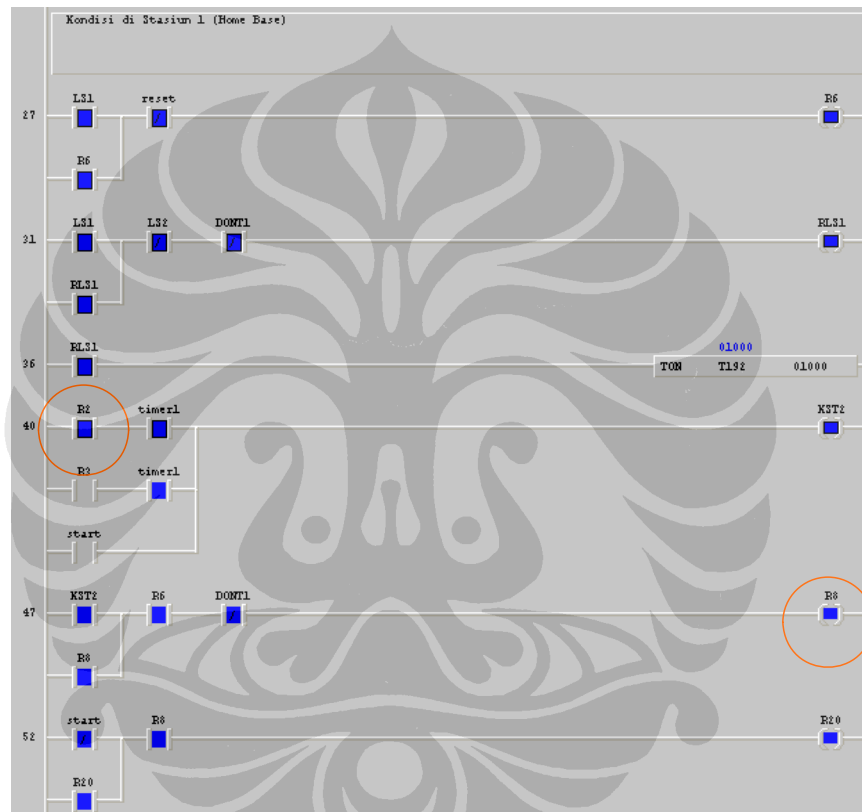
Gambar 4.8. Kondisi awal program di stasiun1

4.3.1 Tram Di Stasiun1 (Skenario Pertama)

Pada kondisi *tram* berada di stasiun1 terdapat beberapa kemungkinan yang dapat terjadi, diantaranya adalah:

- a. Terdapat panggilan dari stasiun2

Ketika terdapat panggilan dari stasiun2 (PBST2/R2 ON) maka secara otomatis *tram* akan bergerak dari stasiun1 menuju stasiun2 (koil R8 aktif) seperti ditunjukkan pada **Gambar 4.9**, dimana koil R8 merupakan salah satu koil yang digunakan untuk menjalankan tram.

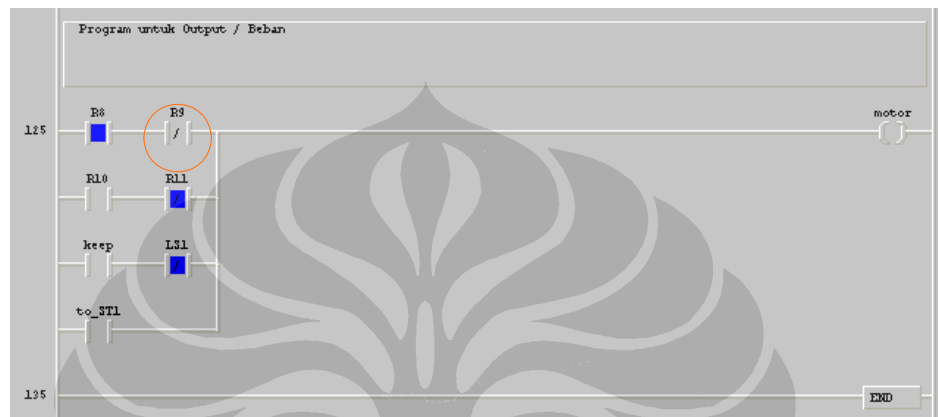


Gambar 4.9. Kondisi terdapat panggilan dari stasiun2

Akan tetapi apabila tombol STOP atau STOP1 (DONT1) diaktifkan, dikarenakan masih terdapat penumpang yang akan masuk kedalam *tram* serta *tram* masih berada di stasiun1 (LS1 ON), maka *tram* berhenti selama 10 detik pada stasiun1, baru setelah itu *tram* bergerak menuju stasiun2. Tombol STOP dan STOP1 dapat digunakan kembali untuk me-reset timer untuk menghitung kembali selama 10 detik pada stasiun1 untuk mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun1 (LS1 ON).

Apabila *tram* sudah meluncur dari stasiun1 maka tombol STOP dan STOP1 tidak dapat digunakan untuk menghentikan *tram*.

Ketika *tram* sudah tiba di stasiun2 (**Gambar 4.10**) LS2 ON dan akan mengaktifkan koil R9, secara otomatis *tram* akan berhenti pada stasiun2. Apabila tidak terdapat panggilan dari stasiun lain ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun2. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.



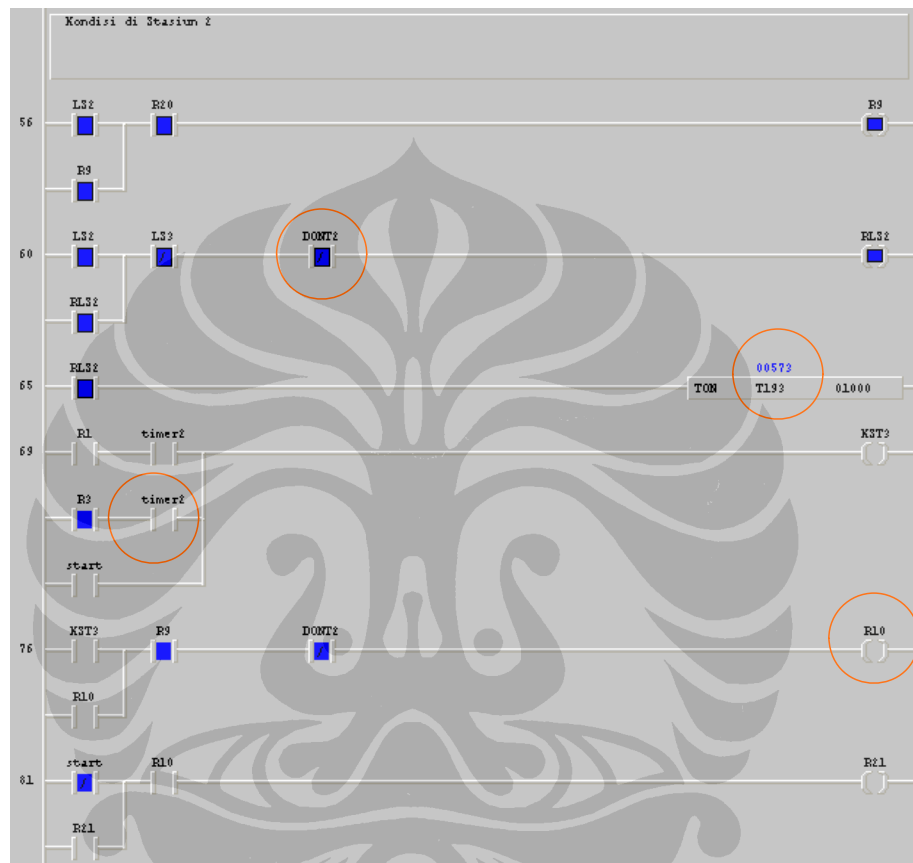
Gambar 4.10. Kondisi *tram* berhenti stasiun2

b. Terdapat panggilan dari stasiun3

Ketika terdapat panggilan dari stasiun3 (PBST3 ON) maka secara otomatis *tram* akan bergerak dari stasiun1 menuju stasiun3 dengan melewati stasiun2 (berhenti di stasiun2) terlebih dahulu. Akan tetapi apabila tombol STOP atau STOP1 (DONT1) diaktifkan dikarenakan masih terdapat penumpang yang akan masuk kedalam *tram* serta *tram* masih berada di stasiun1 (LS1 ON), maka *tram* berhenti selama 10 detik pada stasiun1, baru setelah 10 detik *tram* bergerak menuju stasiun2. Tombol STOP dan STOP1 dapat digunakan kembali untuk me-reset timer menghitung selama 10 detik pada stasiun1 guna mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun1 (LS1 ON).

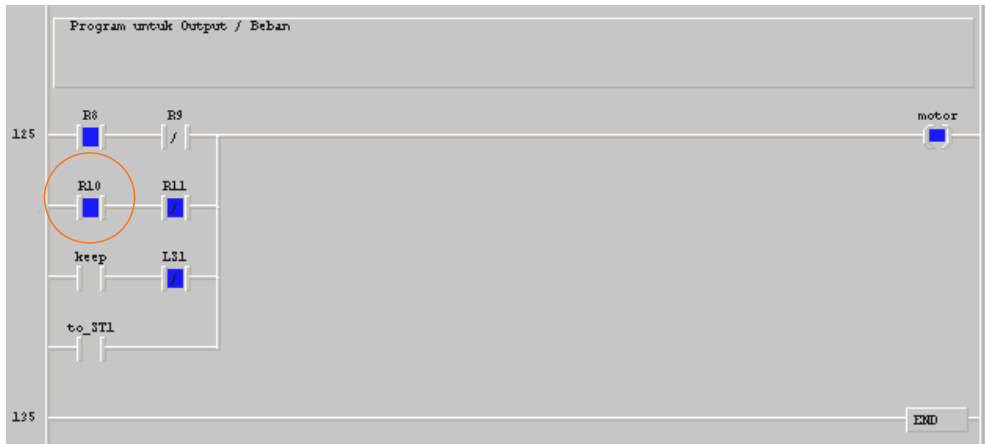
Ketika *tram* sudah tiba di stasiun2 (LS2 ON) maka secara otomatis *tram* akan berhenti pada stasiun2. *Tram* berhenti di stasiun2 selama 10 detik untuk kemudian melanjutkan perjalanan menuju stasiun3. Akan tetapi apabila ternyata tidak ada penumpang yang turun di stasiun2 maka untuk

mempercepat perjalanan serta menghindari waktu 10 detik pemberhentian di stasiun2, penumpang dapat mengaktifkan tombol START untuk menjalankan kembali *tram*. Pemberhentian *tram* di stasiun2 digunakan untuk mengantisipasi kebutuhan penumpang yang berada di stasiun2, karena apabila *tram* langsung menuju stasiun3 maka penumpang di stasiun2 harus menunggu lebih lama untuk menggunakan *tram*.



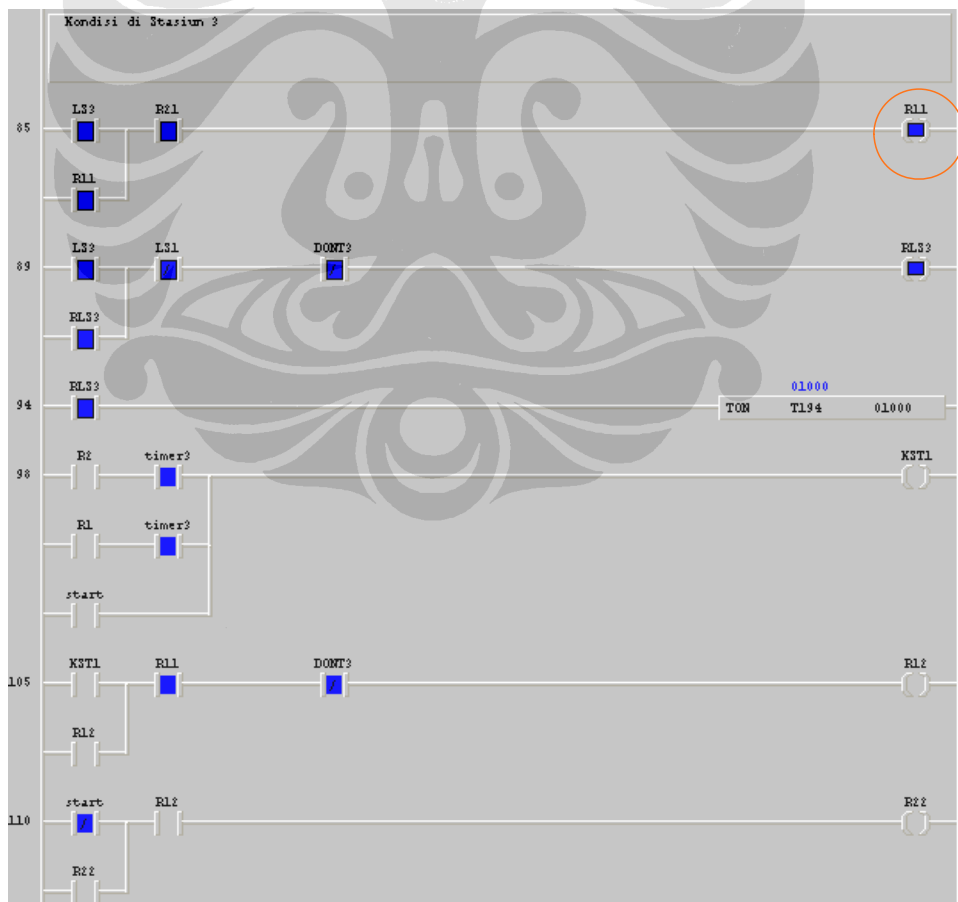
Gambar 4.11. Kondisi *tram* berhenti stasiun2 selama 10 detik

Apabila di stasiun2 masih terdapat penumpang yang akan naik maupun turun, tombol STOP atau STOP2 (DONT2) dapat diaktifkan untuk menghentikan *tram* selama 10 detik pada stasiun2 seperti yang ditunjukkan pada **Gambar 4.11**, setelah 10 detik *tram* bergerak menuju stasiun3. Tombol STOP dan STOP2 dapat digunakan kembali untuk me-reset timer menghitung selama 10 detik pada stasiun2 guna mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun2 (LS2 ON), apabila *tram* tidak berada di stasiun2 maka tombol STOP dan STOP2 tidak dapat digunakan.



Gambar 4.12. Kondisi *tram* menuju stasiun3

Pada **Gambar 4.12** dapat dijelaskan *tram* bergerak menuju stasiun3 dari stasiun2 oleh kontak dari koil R10. Ketika *tram* sudah tiba di stasiun3 (LS3/ R11 ON) maka secara otomatis *tram* akan berhenti pada stasiun3.



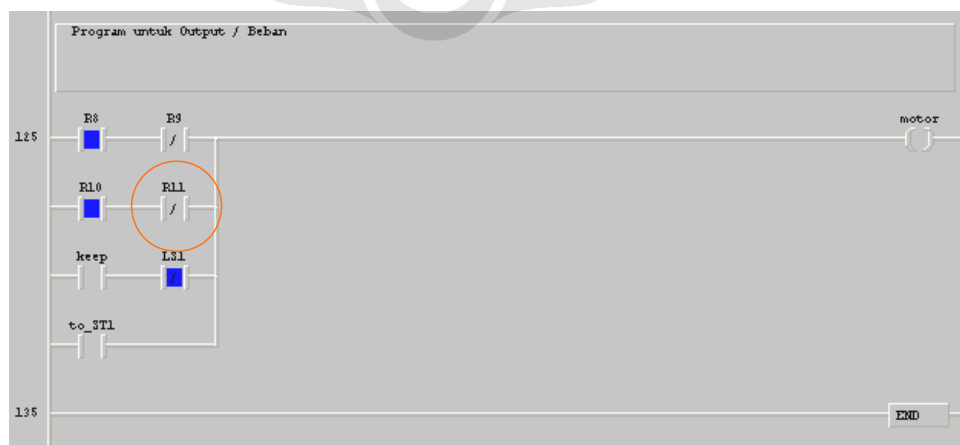
Gambar 4.13. Kondisi program dimana *tram* berada di stasiun3

Apabila tidak terdapat panggilan dari stasiun lain ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun3 seperti yang ditunjukkan pada **Gambar4.13**. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.

c. Terdapat panggilan dari stasiun2 dan stasiun3

Ketika terdapat panggilan dari stasiun2 (PBST2 ON) dan stasiun3 (PBST3 ON) maka secara otomatis *tram* akan bergerak dari stasiun1 menuju stasiun2 dan berhenti di stasiun3 sebagai akhir perjalanan. Secara umum cara kerja pada kondisi ini sama dengan kedua kondisi sebelumnya. *Tram* terlebih dahulu berhenti di stasiun2 selama 10 detik lalu setelah itu akan bergerak menuju stasiun3, apabila masih terdapat penumpang di stasiun2 maka tombol STOP atau STOP2 (DONT2) dapat digunakan untuk menghentikan *tram* di stasiun2 selama 10 detik kembali dengan syarat *tram* masih berada di stasiun2 (LS2 ON). Apabila ternyata kondisi penumpang di stasiun2 telah selesai dan timer masih menghitung, untuk mempercepat dapat digunakan tombol START untuk menjalankan *tram* menuju stasiun3. Setelah kondisi di stasiun2 selesai maka *tram* akan melanjutkan perjalanannya menuju stasiun3.

Ketika *tram* sudah tiba di stasiun3 seperti yang ditunjukkan **Gambar 4.14** (LS3/ R11 ON) maka secara otomatis *tram* akan berhenti pada stasiun3. Apabila tidak terdapat panggilan dari stasiun lain ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun3. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.



Gambar 4.14. Kondisi pemberhentian akhir *tram* di stasiun3

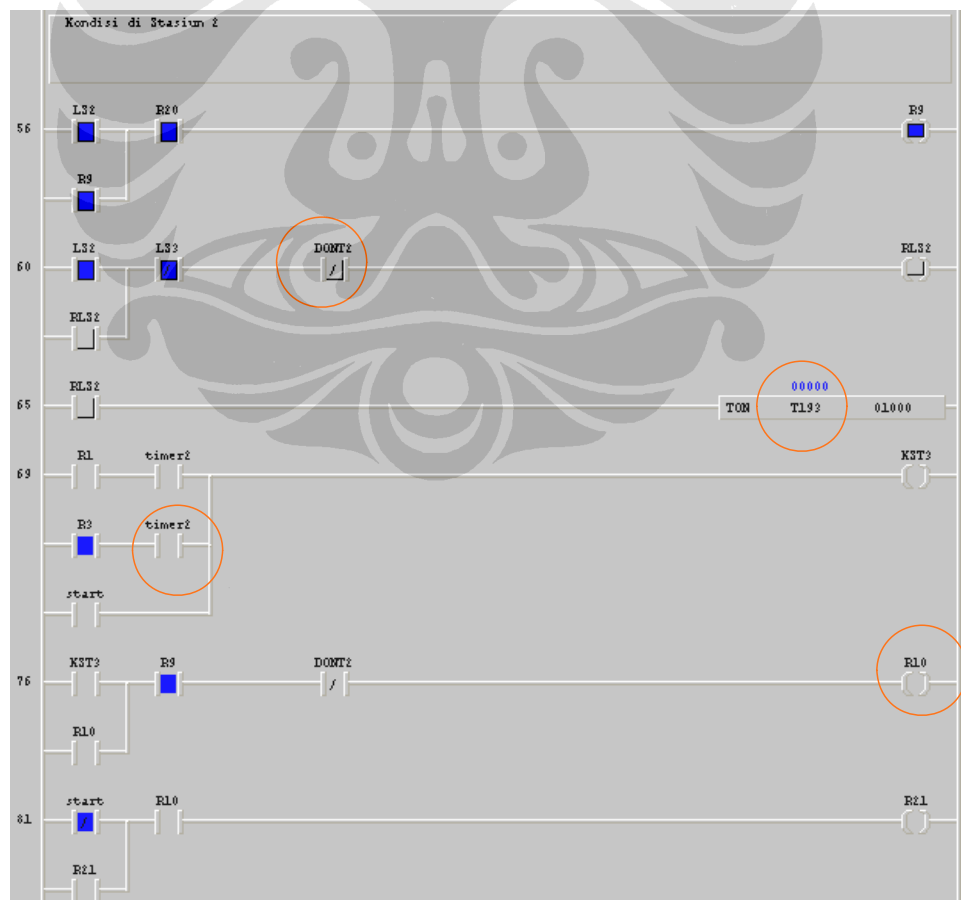
Jika *tram* bergerak meninggalkan stasiun1 dan terdapat panggilan di stasiun1 maka penumpang tersebut harus menunggu terlebih dahulu *tram* yang bergerak menuju stasiun lain untuk kembali menuju stasiun1.

4.3.2 Tram Di Stasiun2 (Skenario Kedua)

Pada kondisi *tram* berada di stasiun2 terdapat beberapa kemungkinan yang dapat terjadi, diantaranya adalah:

a. Terdapat panggilan dari stasiun3

Ketika terdapat panggilan dari stasiun3 (PBST3 ON) maka secara otomatis *tram* akan bergerak dari stasiun2 menuju stasiun3. Akan tetapi apabila tombol STOP atau STOP2 (DONT2) diaktifkan seperti ditunjukkan oleh **Gambar 4.15**, dikarenakan masih terdapat penumpang yang akan menggunakan *tram*, maka *tram* berhenti selama 10 detik pada stasiun2, baru setelah 10 detik *tram* bergerak menuju stasiun3.



Gambar 4.15. Kondisi program dimana tombol STOP aktif

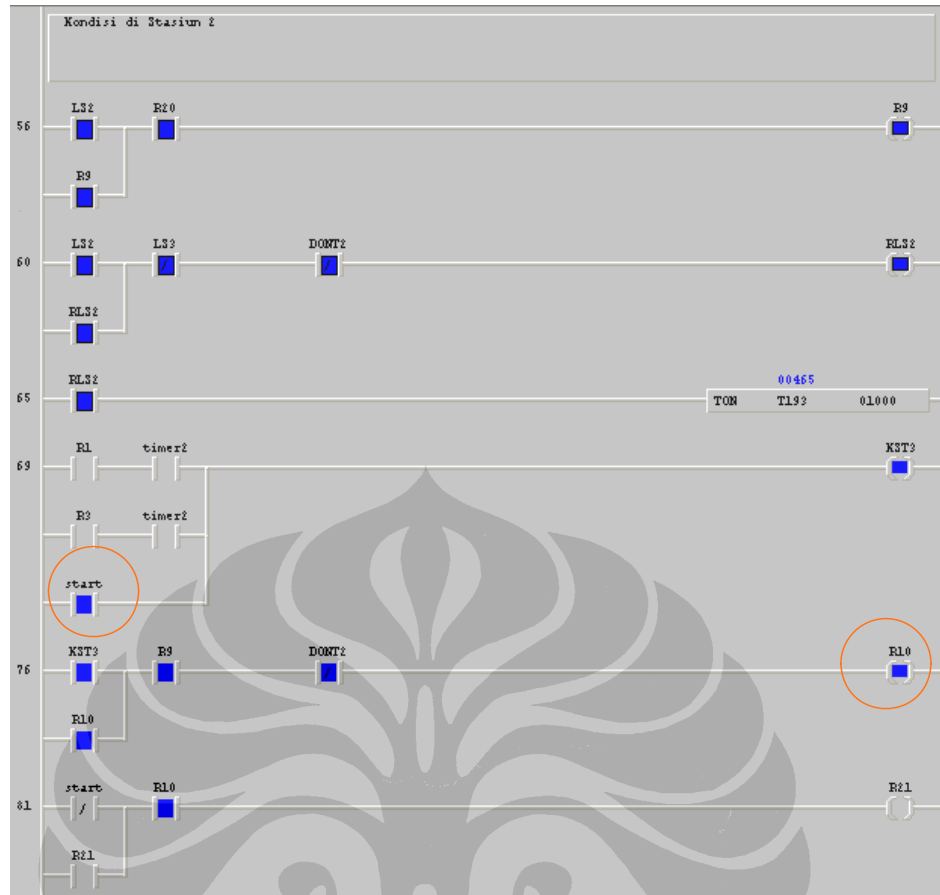
Tombol STOP dan STOP2 dapat digunakan kembali untuk me-*reset timer* menghitung selama 10 detik pada stasiun2 guna mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun2 (LS2 ON).

Ketika *tram* sudah tiba di stasiun3 (LS3 ON) maka secara otomatis *tram* akan berhenti pada stasiun3. Apabila tidak terdapat panggilan dari stasiun lain ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun3. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.

b. Terdapat panggilan dari stasiun1

Ketika terdapat panggilan dari stasiun1 (PBST1 ON) maka secara otomatis *tram* akan bergerak dari stasiun2 menuju stasiun1 dengan melewati stasiun3 (berhenti di stasiun3) terlebih dahulu. Akan tetapi apabila tombol STOP dan STOP2 diaktifkan dikarenakan masih terdapat penumpang yang akan masuk kedalam *tram* serta *tram* masih berada di stasiun2 (LS2 ON), maka *tram* berhenti selama 10 detik pada stasiun2, baru setelah 10 detik *tram* bergerak menuju stasiun3. Tombol STOP dan STOP2 dapat digunakan kembali untuk me-*reset timer* menghitung selama 10 detik pada stasiun2 guna mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun2 (LS2 ON).

Ketika *tram* sudah tiba di stasiun3 (LS3 ON) maka secara otomatis *tram* akan berhenti pada stasiun3. *Tram* berhenti di stasiun3 selama 10 detik untuk kemudian melanjutkan perjalanan menuju stasiun1. Akan tetapi apabila ternyata tidak ada penumpang yang turun di stasiun3 maka untuk mempercepat perjalanan serta menghindari waktu 10 detik pemberhentian di stasiun3, penumpang dapat mengaktifkan tombol START untuk menjalankan kembali *tram* seperti yang ditunjukkan pada **Gambar 4.16**. Kontak tombol START langsung terhubung dengan koil KST3 yang akan mengaktifkan koil R10 yang berfungsi untuk mengaktifkan koil *output* motor. Pemberhentian *tram* di stasiun3 digunakan untuk mengantisipasi kebutuhan penumpang yang berada di stasiun3, karena apabila *tram* langsung menuju stasiun1 maka penumpang di stasiun3 harus menunggu lebih lama untuk menggunakan *tram*.



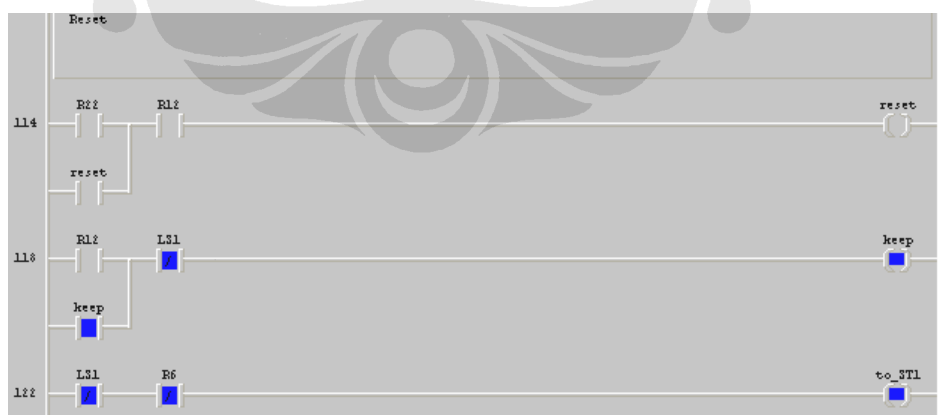
Gambar 4.16. Kondisi program dimana tombol START aktif

Apabila di stasiun3 masih terdapat penumpang yang akan naik maupun turun, tombol STOP dan STOP3 dapat diaktifkan untuk menghentikan *tram* selama 10 detik pada stasiun3, setelah 10 detik *tram* bergerak menuju stasiun1. Tombol STOP dan STOP3 dapat digunakan kembali untuk me-reset timer menghitung selama 10 detik pada stasiun3 guna mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun3 (LS3 ON).

Ketika *tram* sudah tiba di stasiun1 (LS1 ON) maka secara otomatis *tram* akan berhenti pada stasiun1. Apabila tidak terdapat panggilan dari stasiun lain ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun1. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.

- c. Terdapat panggilan dari stasiun3 dan stasiun1

Ketika terdapat panggilan dari stasiun3 (PBST3 ON) dan stasiun1 (PBST1 ON) maka secara otomatis *tram* akan bergerak dari stasiun2 menuju stasiun3 dan berhenti di stasiun1 sebagai akhir perjalanan. Secara umum cara kerja pada kondisi ini sama dengan kedua kondisi sebelumnya. *Tram* terlebih dahulu berhenti di stasiun3 selama 10 detik lalu setelah itu akan bergerak menuju stasiun1, apabila masih terdapat penumpang di stasiun3 maka tombol STOP dan STOP3 dapat digunakan untuk menghentikan *tram* di stasiun3 selama 10 detik kembali dengan syarat *tram* masih berada di stasiun3 (LS3 ON). Apabila ternyata kondisi penumpang di stasiun3 telah selesai dan timer masih menghitung, untuk mempercepat dapat digunakan tombol START untuk menjalankan *tram* menuju stasiun1. Setelah kondisi di stasiun3 selesai maka *tram* akan melanjutkan perjalanannya menuju stasiun1. Secara program proses kembalinya *tram* menuju stasiun1 merupakan sebuah siklus, oleh karena pada perancangan program dengan menggunakan metode Y-Map maka dibutuhkan satu bagian program yang fungsinya me-*reset* ke keadaan awal yaitu ketika *tram* berada di stasiun1. Pada **Gambar 4.17** ditunjukkan bagian program *reset* dimana ketika *tram* bergerak menuju stasiun1 maka kontak R22 akan mengaktifkan koil reset. Kontak reset akan melepaskan kunci (SHR) pada semua program kondisi stasiun, sehingga program kembali ke awal.



Gambar 4.17. Kondisi menuju stasiun1 dari stasiun3 (*reset* program)

Ketika *tram* sudah tiba di stasiun1 (LS1 ON) maka secara otomatis *tram* akan berhenti pada stasiun1. Apabila tidak terdapat panggilan dari stasiun lain

ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun1. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.

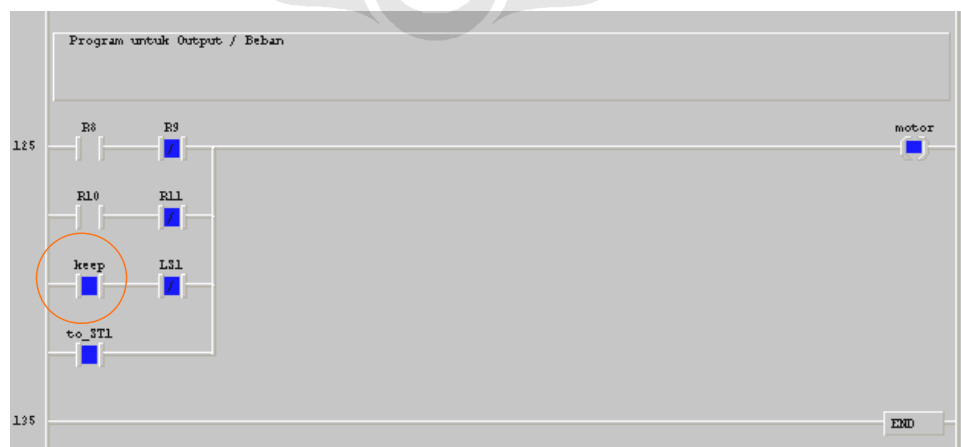
Jika *tram* bergerak meninggalkan stasiun2 dan terdapat panggilan di stasiun2 maka penumpang tersebut harus menunggu terlebih dahulu *tram* yang bergerak menuju stasiun lain untuk kembali menuju stasiun2.

4.3.3 Tram Di Stasiun3 (Skenario Ketiga)

Pada kondisi *tram* berada di stasiun3 terdapat beberapa kemungkinan yang dapat terjadi, diantaranya adalah:

- a. Terdapat panggilan dari stasiun1

Ketika terdapat panggilan dari stasiun1 (PBST1 ON) maka secara otomatis *tram* akan bergerak dari stasiun3 menuju stasiun1 seperti yang ditunjukkan oleh **Gambar 4.18**. Kontak keep adalah kontak yang digunakan untuk mengaktifkan koil *output* motor. Akan tetapi apabila tombol STOP dan STOP3 diaktifkan, dikarenakan masih terdapat penumpang yang akan masuk kedalam *tram* serta *tram* masih berada di stasiun3 (LS3 ON), maka *tram* berhenti selama 10 detik pada stasiun3, baru setelah 10 detik *tram* bergerak menuju stasiun1. Tombol STOP dan STOP3 dapat digunakan kembali untuk *me-reset timer* menghitung selama 10 detik pada stasiun3 guna mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun3 (LS3 ON).



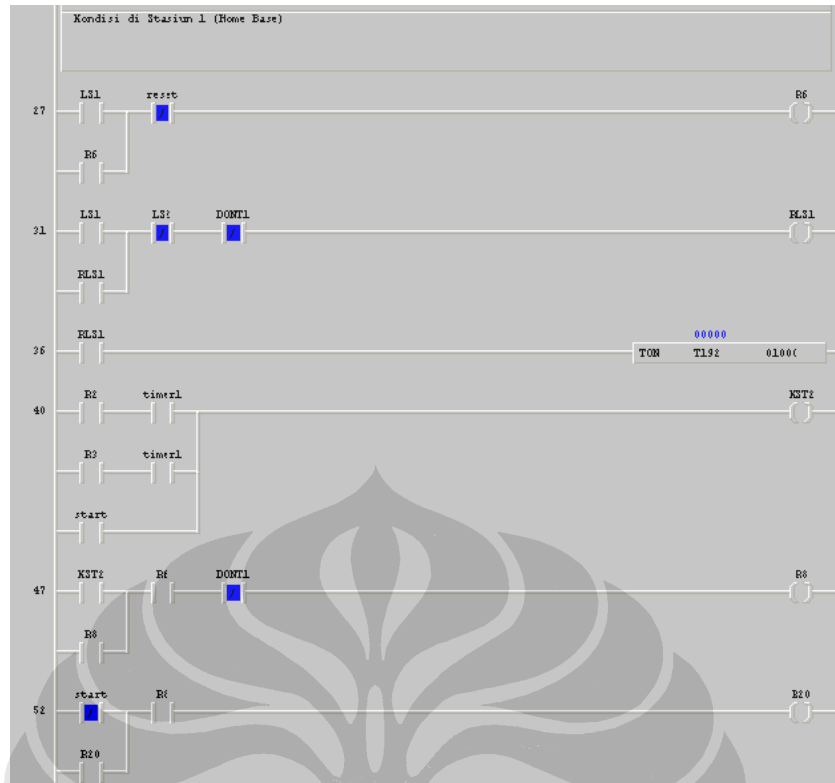
Gambar 4.18. Kondisi *tram* menuju stasiun1 dari stasiun3

Ketika *tram* sudah tiba di stasiun1 (LS1 ON) maka secara otomatis *tram* akan berhenti pada stasiun1. Apabila tidak terdapat panggilan dari stasiun lain ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun1. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.

b. Terdapat panggilan dari stasiun2

Ketika terdapat panggilan dari stasiun2 (PBST2 ON) maka secara otomatis *tram* akan bergerak dari stasiun3 menuju stasiun2 dengan melewati stasiun1 (berhenti di stasiun1) terlebih dahulu. Akan tetapi apabila tombol STOP dan STOP3 diaktifkan dikarenakan masih terdapat penumpang yang akan masuk kedalam *tram* serta *tram* masih berada di stasiun3 (LS3 ON), maka *tram* berhenti selama 10 detik pada stasiun3, baru setelah 10 detik *tram* bergerak menuju stasiun1. Tombol STOP dan STOP3 dapat digunakan kembali untuk me-*reset timer* menghitung selama 10 detik pada stasiun3 guna mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun3 (LS3 ON).

Ketika *tram* sudah tiba di stasiun1 (LS1 ON) maka secara otomatis *tram* akan berhenti pada stasiun1. Proses dimana *tram* kembali menuju stasiun1 adalah sebuah siklus yang menggunakan suatu program reset untuk mengembalikan seluruh bagian program kembali ke kondisi awal. Pada **Gambar 4.19** dapat dilihat kondisi program di stasiun1 setelah program reset diaktifkan, kontak reset digunakan untuk melepaskan penguncian pada koil R8 dan R20. *Tram* berhenti di stasiun1 selama 10 detik untuk kemudian melanjutkan perjalanan menuju stasiun2. Akan tetapi apabila ternyata tidak ada penumpang yang turun di stasiun1 maka untuk mempercepat perjalanan serta menghindari waktu 10 detik pemberhentian di stasiun1, penumpang dapat mengaktifkan tombol START untuk menjalankan kembali *tram*. Pemberhentian *tram* di stasiun1 digunakan untuk mengantisipasi kebutuhan penumpang yang berada di stasiun1, karena apabila *tram* langsung menuju stasiun2 maka penumpang di stasiun1 harus menunggu lebih lama untuk menggunakan *tram*.



Gambar 4.19. Kondisi di stasiun1 setelah program *reset* aktif

Apabila di stasiun1 masih terdapat penumpang yang akan naik maupun turun, tombol STOP dan STOP1 dapat diaktifkan untuk menghentikan *tram* selama 10 detik pada stasiun1, setelah 10 detik *tram* bergerak menuju stasiun2. Tombol STOP dan STOP1 dapat digunakan kembali untuk me-*reset* timer menghitung selama 10 detik pada stasiun1 guna mengantisipasi kondisi penumpang, tetapi dengan syarat *tram* masih berada di stasiun1 (LS1 ON).

Ketika *tram* sudah tiba di stasiun2 (LS2 ON) maka secara otomatis *tram* akan berhenti pada stasiun2. Apabila tidak terdapat panggilan dari stasiun lain ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun2. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.

- c. Terdapat panggilan dari stasiun1 dan stasiun2

Ketika terdapat panggilan dari stasiun1 (PBST1 ON) dan stasiun2 (PBST2 ON) maka secara otomatis *tram* akan bergerak dari stasiun3 menuju stasiun1 dan berhenti di stasiun2 sebagai akhir perjalanan. Secara umum cara

kerja pada kondisi ini sama dengan kedua kondisi sebelumnya. *Tram* terlebih dahulu berhenti di stasiun1 selama 10 detik lalu setelah itu akan bergerak menuju stasiun2, apabila masih terdapat penumpang di stasiun1 maka tombol STOP dan STOP1 dapat digunakan untuk *me-reset timer* menghitung selama 10 detik pada stasiun1 dengan syarat *tram* masih berada di stasiun1 (LS1 ON). Apabila ternyata kondisi penumpang di stasiun1 telah selesai dan timer masih menghitung, untuk mempercepat dapat digunakan tombol START untuk menjalankan *tram* menuju stasiun2. Setelah kondisi di stasiun1 selesai maka *tram* akan melanjutkan perjalanannya menuju stasiun2.

Ketika *tram* sudah tiba di stasiun2 (LS2 ON) maka secara otomatis *tram* akan berhenti pada stasiun2. Apabila tidak terdapat panggilan dari stasiun lain ataupun tombol START tidak diaktifkan maka *tram* akan tetap berada di stasiun2. *Tram* dapat dijalankan menuju stasiun lain dengan menggunakan tombol START.

Jika *tram* bergerak meninggalkan stasiun3 dan terdapat panggilan di stasiun3 maka penumpang tersebut harus menunggu terlebih dahulu *tram* yang bergerak menuju stasiun lain untuk kembali menuju stasiun3.

Dari keseluruhan ujicoba yang telah dilakukan terdapat masalah yang timbul yaitu, ketika tombol STOP, STOP1, STOP2, STOP3 digunakan. Hal tersebut dikarenakan penggunaan keempat *input* diatas memiliki syarat dimana salah satu kondisi sensor pada stasiun (LS1, LS2, LS3) harus dalam kondisi ON (menyatakan *tram* berada di stasiun). Sementara perubahan kondisi sensor dari kondisi ON ke OFF dipengaruhi pergerakan *tram* yang sangat cepat dikarenakan tidak terdapat suatu kontrol kecepatan motor pada *tram*. Kondisi *tram* yang bergerak cepat meninggalkan stasiun menyebabkan sensor menjadi OFF sehingga keempat tombol tersebut tidak dapat digunakan. Permasalahan diatas dapat diatasi dengan jalan menambah fungsi *timer* (secara program) sebagai indikator *tram* akan bergerak, ataupun menggabungkan dengan sistem lain contohnya kontrol pintu *tram* dan pintu stasiun, pengaturan kecepatan motor (*electric drive*) serta konstruksi mekanis dari sensor itu sendiri.

BAB V

KESIMPULAN

5.1 KESIMPULAN

Berdasarkan analisa dan hasil ujicoba program PLC terhadap minatur *tram* dapat diambil kesimpulan dari pembuatan tugas akhir ini yaitu :

1. Telah berhasil dirancang program *ladder* PLC untuk mengendalikan panggilan dan pemberhentian *tram* secara otomatis pada tiap-tiap stasiun. Program *ladder* yang dibuat terdiri dari 135 langkah.
2. Telah berhasil dibangun miniatur kereta listrik (*tram*) dengan jumlah stasiun tiga buah.
3. Telah berhasil diujicobakan program (*ladder* PLC) panggilan *tram* secara otomatis pada stasiun, diaplikasikan terhadap miniatur yang dibuat.

DAFTAR ACUAN

- [1] Permadi Hudoyo. “Pemrograman PLC Pada Modul Latih Inverter Berbasis PLC.” Tugas Akhir, Program Diploma Teknik Listrik PNJ, Depok, 2005, hal. 8.
- [2] LG Industrial System, *LG Programmable Logic Controller Instruction & Programming*. (User’s Manual LG-PLC MASTER-K, 2003), Ch. 2-7 – 2-9.
- [3] Achmad Fauzi, *Modul Praktikum Laboratorium PLC* (Teknik Listrik, Politeknik Negeri Jakarta, 2004), hal. 9-13.



DAFTAR PUSTAKA

Fauzi, Achmad. *Modul Praktikum Laboratorium PLC* (Teknik Listrik, Politeknik Negeri Jakarta, 2004), hal. 9-13.

<http://en.wikipedia.org/wiki/Tram>.

Hudoyo, Permadi. "Pemrograman PLC Pada Modul Latih Inverter Berbasis PLC." Tugas Akhir, Program Diploma Teknik Listrik PNJ, Depok, 2005, hal. 8.

Isnandar, Dadang. "Kontrol Motor Induksi Pada Modul Latih Inverter Berbasis PLC." Tugas Akhir, Program Diploma Teknik Listrik PNJ, Depok, 2005, hal. 5-13.

LG Industrial System, *LG Programmable Logic Controller Instruction & Programming*. (User's Manual LG-PLC MASTER-K, 2003), Ch. 2-7 – 2-9.

LG Industrial System, *MASTER-K 120S Programmable Logic Controller*. (Catalogs LG-PLC MASTER-K 120S, 2003), hal. 8-12.

Sujatmoko, MN (2000). *Dasar-dasar Control Component dan SYSMAC*. Diakses 25 Mei 2008 dari wordpress.
<http://ninafkoe.files.wordpress.com/2008/04/basic-plc.pdf>

LAMPIRAN

a)

Specification	Model	Main unit			
		K7M-DR20U	K7M-DR30U	K7M-DR40U	K7M-DR60U
		K7M-DRT20U	K7M-DRT30U	K7M-DRT40U	K7M-DRT60U
Input points		12 points	18 points	24 points	36 points
Insulation method	Photocoupler				
Rated input voltage	DC 24V				
Rated input current	7mA				
Operating voltage range	DC 20.4~28.8V (ripple rate < 5%)				
Max. simultaneous input points	100% Simultaneous on				
On Voltage/On current	DC 19V or higher / 5.7mA or higher				
Off Voltage/Off current	DC 6V or lower / 1.8mA or lower				
Input impedance	Approximate 3.3k Ω				
Response time	Off \rightarrow On	0, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000ms (default: 10ms), Set by User			
	On \rightarrow Off	0, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000ms (default: 10ms)			
Common terminal		12 points/COM	18 points/COM	12 points/COM	18 points/COM
Operating indicator	LED				

b)

Specification	Model	Main unit			
		K7M-DR20U	K7M-DR30U	K7M-DR40U	K7M-DR60U
Output points		8	12	16	24
Insulation method	Relay insulation				
Rated load voltage/current	DC 24V/2A (Resistor load), AC 220V/2A (COS ϕ = 1)/1point 5A/COM				
Min. load voltage/current	DC 5V/1mA				
Max. load voltage/current	AC 250V, DC 110V				
Off leakage current	0.1mA (AC 220V, 60Hz)				
Max. on/off frequency	1200/hr				
Surge absorber	None				
Service life	Mechanical	20 million times or more			
	Electrical	Rated load voltage/current: 100,000 or more AC 200V/1.5A, AC 240V/1A (COS ϕ = 0.7) 100,000 or more AC 200V/1A, AC 240V/0.5A (COS ϕ = 0.35) 100,000 or more DC 24V/1A, DC 100V/0.1A (L/R = 7ms) 100,000 or more.			
Response time	Off \rightarrow On	10ms or less			
	On \rightarrow Off	12ms or less			
Operation indication	LED				

a) Spesifikasi Input PLC MASTER-K120S Tipe K7M-DR30U

a) Spesifikasi Output PLC MASTER-K120S Tipe K7M-DR30U



UNIVERSITAS INDONESIA

DEPARTEMEN ELEKTRO

FTUI

2008