

***Prototype Penyimpanan Benda Bertingkat Otomatis
dengan Pengaman Password dan RFID Berbasis PLC
Menggunakan Human Machine Interface***

(Suatu Studi Eksperimen di Laboratorium PLC)

MUHAMMAD RIZKI ATHOILLAH
5115116377



Skripsi ini ditulis untuk memenuhi sebagian persyaratan dalam memperoleh gelar Sarjana Pendidikan.

**PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI JAKARTA
2016**

ABSTRAK

MUHAMMAD RIZKI ATHOILLAH, *Prototype Penyimpanan Benda Bertingkat Otomatis dengan Pengaman Password dan RFID Berbasis PLC Menggunakan Human Machine Interface.* Program Studi Pendidikan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta, 2016, Pembimbing Syufrijal, S.T., M.T. dan Nur Hanifah Yuninda, S.T., M.T.

Penelitian ini bertujuan membuat *prototype* penyimpanan benda bertingkat otomatis menggunakan *Programmable Logic Controller (PLC)* CJ1M dengan pengaman *password* dan RFID serta *interface visual basic*. Penelitian menggunakan metode eksperimen dan dilaksanakan di laboratorium bengkel mekanik Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta pada bulan Juli 2015 sampai Desember 2015. *Prototype* menggunakan PLC sebagai pusat penggerak komponen mekaniknya yang diprogram dengan kode *mnemonic* atau *ladder diagram*. Sedangkan untuk sistem keamanan diatur oleh *Radio Frequency Identification (RFID) kit* dan *software* yang dibuat pada *visual basic*. *Visual basic* juga berfungsi sebagai *interface prototype*. Sinyal *input* PLC bersumber dari 12 *limit switch*, 1 sensor infra merah dan tombol pada *visual basic*. *Limit Switch* digunakan untuk membatasi pergerakan pemindah benda. Sensor infra merah berfungsi untuk mendeteksi apakah pada pemindah benda terdapat benda. Sedangkan tombol pada *visual basic* berfungsi sebagai pemilih benda akan disimpan atau diambil. Peralatan *output* terdiri dari 9 buah relai yang terhubung dengan motor DC *axis X*, motor DC *axis Y*, motor DC *axis Z*, motor DC pintu depan serta *doorlock*. 4 buah relai digunakan untuk mengatur daya pada 4 buah motor DC. 4 buah relai digunakan untuk mengatur arah putaran pada 4 buah motor DC. Dan 1 buah relai digunakan untuk mengatur daya pada *doorlock* yang membuat *doorlock* menjadi terbuka atau terkunci. Dari hasil penelitian *prototype* penyimpanan benda bertingkat otomatis dengan pengaman *password* dan RFID berbasis PLC menggunakan *human machine interface*, alat dan sistem dapat bekerja dengan baik, sesuai dengan yang dirancang dan dapat digunakan baik ketika menyimpan dan mengambil benda.

Kata kunci : Penyimpanan Benda Bertingkat, PLC, RFID, *Visual Basic*.

ABSTRACT

MUHAMMAD RIZKI ATHOILLAH, Prototype of PLC Based Automatic Multilevel Storage with Password and RFID Security Using Human Machine Interface. Study Program Electrical Engineering, Faculty of Engineering, State University of Jakarta, in 2016, Advisors Syufrijal, S.T., M.T. and Nur Hanifah Yuninda, S.T., M.T.

This research is intended to make a prototype automatic multilevel storage using Programmable Logic Controller (PLC) CJ1M, password and RFID security system and visual basic interface. This study used experimental method and was held at electronic mechanic laboratory, faculty of engineering, State University of Jakarta on July 2015 until December 2015. This prototype uses Programmable Logic Controller (PLC) as the main mechanical component mover which can be programmed using mnemonic code or ladder diagram. As for security system, it is managed by (Radio Frequency Identification) RFID kit and software made with visual basic. The software made with visual basic is also worked as an interface prototype. PLC input signal is sourced from 12 limit switch, 1 infrared sensor and button on visual basic. Limit Switch is used to limit the degree of movement from the object mover. Infrared sensor works to detect if there is something on the object mover. Whereas the button on visual basic works to choose if the object is to be stored or taken. Output kit consist of 9 relay which is linked by axis X DC motor, axis Y DC motor, axis Z DC motor, front door DC motor and doorlock. 4 relay is used to manage the power on the 4 DC motor. 4 relay is used to manage the course of rotation on the 4 DC motor. And 1 relay is used to manage the power on the doorlock which unlock and lock the doorlock. From the research of the Prototype of PLC Based Automatic Multilevel Storage with Password and RFID Security using human machine interface, the tool and system work appropriately, corresponding with the design and can be used well when storing and taking object.

Keywords:, Multilevel Storage, PLC, RFID, Visual Basic.

LEMBAR PENGESAHAN

NAMA DOSEN	TANDA TANGAN	TANGGAL
Syufrijal, S.T., M.T. (Dosen Pembimbing I)		26 / 1 - 2016
Nur Hanifah Yuninda, S.T., M.T. (Dosen Pembimbing II)		26 / 01 - 2016

PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
Drs. Irzan Zakir, M.Pd. (Ketua Penguji)		25 / 1 / 16
Massus Subekti, S.Pd., M.T. (Sekretaris Penguji)		25 / 1 / 2016
Moch. Djaohar, M.Sc. (Dosen Ahli)		25 / 1 / 2016

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, Januari 2016

Yang membuat pernyataan



Muhammad Rizki Athoillah

5115116377

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan rahmat, karunia dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “*Prototype Penyimpanan Benda Bertingkat Otomatis dengan Pengaman Password dan RFID Berbasis PLC Menggunakan Human Machine Interface*”. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Pendidikan Teknik Elektro pada Fakultas Teknik, Universitas Negeri Jakarta.

Penulis menyadari bahwa skripsi ini tidaklah dapat terwujud dengan baik tanpa adanya bimbingan, dorongan, saran-saran dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Kedua orang tua, Bapak Dedy Dwidjaya dan Ibu Jojoh Johariyah yang senantiasa memberikan dukungan dan doa hingga penulis dapat menyelesaikan pendidikan di Universitas Negeri Jakarta.
2. Bapak Massus Subekti, S.Pd., M.T., selaku Ketua Program Studi Pendidikan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.
3. Bapak Drs. Readysal Monantun, M.M., selaku Pembimbing Akademik Program Studi Pendidikan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.
4. Bapak Syufrijal, S.T., M.T. dan Ibu Nur Hanifah Yuninda, S.T., M.T., selaku dosen pembimbing yang penuh kesabaran dan kepercayaan dalam membimbing dan memberi semangat kepada penulis hingga selesainya skripsi ini.

5. Seluruh dosen Universitas Negeri Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang berguna.
6. Rekan-rekan Mahasiswa Universitas Negeri Jakarta khususnya kelas Non Reguler angkatan 2011 Program Studi Pendidikan Teknik Elektro selaku teman dan sahabat yang selalu memberikan motivasi.
7. Serta semua pihak yang belum penulis sebutkan dalam membantu penyelesaian skripsi ini.

Semoga Allah SWT membalas semua kebaikan semua pihak yang telah membantu. Penulis menyadari bahwa skripsi ini jauh dari kesempurnaan, karena penulis menyadari bahwa tidak ada gading yang tak retak. Untuk itu penulis mohon maaf apabila terdapat kekurangan dan kesalahan baik dari isi maupun tulisan. Semoga skripsi ini bermanfaat bagi yang membacanya dan semua pihak yang terkait.

Jakarta, Januari 2016

Penulis

Muhammad Rizki Athoillah
5115116377

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
ABSTRAK	ii
ABSTRACT	iii
LEMBAR PENGESAHAN	iv
HALAMAN PERNYATAAN	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah	3
1.3. Pembatasan Masalah	3
1.4. Perumusan Masalah.....	4
1.5. Tujuan Penelitian	4
1.6. Manfaat Penelitian	5
BAB II KAJIAN TEORITIS, DAN KERANGKA BERFIKIR.	7
2.1. <i>Programmable Logic Controller</i> (PLC).....	7
2.1.1. Pengertian PLC	7
2.1.2. Komponen Utama PLC.....	9
2.1.2.1. <i>Central Processing Unit</i> (CPU).....	11
2.1.2.2. Modul <i>Input</i> dan <i>Output</i> (I/O).....	13
2.1.2.3. <i>Power Supply</i>	15
2.1.2.4. Perangkat Pemrograman	15
2.1.3. Pengalamatan I/O PLC	18
2.1.4. Instruksi Dalam Pemrograman PLC	20
2.1.4.1. Instruksi-instruksi Dasar	20
2.1.4.2. Instruksi <i>KEEP</i>	22
2.1.4.4. Instruksi <i>Timer</i>	22
2.1.4.5. Instruksi <i>Counter</i>	24
2.1.5. Pengaturan <i>Hardware</i> PLC.....	25
2.2. <i>Visual Basic</i>	30
2.2.1. Kelebihan <i>Visual Basic</i>	30
2.2.2. Bagian-bagian <i>Visual Basic (Integrated Development Environment)</i>	31
2.2.3. Kontrol Program <i>Visual Basic</i>	36
2.2.3.1. <i>If ...Then</i>	36
2.2.3.2. <i>If...Then...Else</i>	37
2.2.3.3. <i>Select...Case</i>	38
2.2.4. Antarmuka <i>Intelligent Electronic Device</i> (IED) melalui <i>Visual Basic</i>	39
2.3. Konfigurasi Manipulator Robot Kartesian	44
2.4. Sensor Cahaya.....	44

2.5. <i>Radio Frequency Identification (RFID)</i>	46
2.5.1. <i>RFID tag</i>	46
2.5.2. <i>RFID reader</i>	47
2.6. Motor DC Magnet Permanen.....	48
2.7. Gear Box (Hubungan Roda-roda).....	50
2.7.1. Dua Roda dengan Poros yang Sama.....	50
2.7.2. Dua Roda Dihubungkan Satu Rantai.....	52
2.7.3. Roda Saling Bersinggungan.....	53
2.8. Saklar Pembatas (<i>Limit Switch</i>)	55
2.9. Relai	56
2.10. <i>Solenoid Doorlock</i>	57
2.11. Kerangka Berfikir.....	58
BAB III METODOLOGI PENELITIAN	61
3.1. Tempat dan Waktu Penelitian.....	61
3.2. Metode Penelitian.....	61
3.3. Rancangan Penelitian	61
3.3.1. Perancangan Mekanik Penyimpanan Benda Bertingkat Otomatis.....	61
3.3.2. Perancangan sensor infra merah.....	64
3.3.3. Perancangan <i>wiring</i>	67
3.3.4. Perancangan Program	70
3.3.4.1. Deskripsi Kerja Sistem	70
3.3.4.2. Penetapan Alamat <i>Input</i> dan Alamat <i>Output</i> PLC.....	76
3.3.4.3. Program PLC.....	79
3.3.4.4. Program <i>visual basic</i>	79
3.4. Instrumen Penelitian.....	80
3.4.1. Pengujian Sensor Infra Merah.....	81
3.4.2. Pengujian RFID.....	81
3.4.3. Pengujian tegangan <i>Prototype</i>	82
3.4.4. Pengujian Sistem Pengaman	84
3.4.5. Pengujian Penyimpanan dan Pengambilan <i>Software Visual Basic</i>	84
3.4.6. Pengujian Waktu Kerja.....	86
BAB IV HASIL PENELITIAN	87
4.1. Hasil Pengujian Alat.....	87
4.1.1. Hasil Pengujian Sensor Infra Merah.....	87
4.1.2. Hasil Pengujian RFID.....	87
4.1.3. Hasil Pengujian Tegangan <i>Prototype</i>	88
4.1.4. Hasil Pengujian Sistem Pengaman	90
4.1.5. Hasil Pengujian Penyimpanan dan Pengambilan <i>Software Visual Basic</i>	91
4.1.6. Hasil Pengujian Waktu Kerja.....	93
4.2. Analisis Hasil Pengujian Alat	94
4.2.1. Analisis Hasil Pengujian Sensor Infra Merah	94
4.2.2. Analisis Hasil Pengujian RFID	95
4.2.3. Analisis Hasil Pengujian Tegangan <i>Prototype</i>	97
4.2.4. Analisis Hasil Pengujian Sistem Pengaman	97

4.2.5. Analisis Hasil Pengujian Penyimpanan dan Pengambilan <i>Software Visual Basic</i>	100
4.2.6. Analisis Hasil Pengujian Waktu Kerja	103
4.3. Kelebihan dan Kekurangan Alat	103
4.3.1. Kelebihan Alat	103
4.3.2. Kekurangan Alat	104
BAB V KESIMPULAN DAN SARAN	105
4.1. Kesimpulan	105
4.2. Saran	106
DAFTAR PUSTAKA	107
LAMPIRAN	109
DAFTAR RIWAYAT HIDUP	160

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Interaksi Komponen-komponen sistem PLC	10
Gambar 2.2. Representasi <i>scan</i> PLC	12
Gambar 2.3. <i>Layout</i> Modul <i>Input</i>	14
Gambar 2.4. <i>Layout</i> modul <i>Output</i>	14
Gambar 2.5. <i>Miniprogrammer</i> Omron.....	16
Gambar 2.6. Konektifitas PC ke PLC.....	17
Gambar 2.7. Pengalamatan I/O CS1/CJ1.....	18
Gambar 2.8. Pengalamatan I/O CS1/CJ1 pada Rak Ekspansi.....	19
Gambar 2.9. Instruksi AND	21
Gambar 2.10. Instruksi OR	21
Gambar 2.11. Simbol <i>Ladder</i> dan Area Data <i>Operand</i> instruksi <i>Keep</i>	22
Gambar 2.12. Contoh Penggunaan Instruksi <i>Keep</i>	22
Gambar 2.13. Simbol <i>Ladder</i> , Nomor, dan Area Data <i>Operand Timer</i>	23
Gambar 2.14. Contoh Penggunaan <i>Timer</i>	23
Gambar 2.15. Diagram Waktu <i>Timer</i>	24
Gambar 2.16. Simbol <i>Ladder</i> , Nomor dan Area Data <i>Operand Counter</i>	24
Gambar 2.17. Contoh Penggunaan <i>Counter</i>	25
Gambar 2.18. Diagram Waktu <i>Counter</i>	25
Gambar 2.19. Perancangan <i>Hardware</i> PLC.....	26
Gambar 2.20. Jendela Awal <i>CX-Programmer</i>	26
Gambar 2.21. <i>Setting</i> awal PLC	27
Gambar 2.22. Tampilan Awal Program.....	27
Gambar 2.23. Tampilan Konfigurasi <i>Hardware</i>	28
Gambar 2.24. Pilihan <i>Hardware</i> Unit PLC.....	28
Gambar 2.25. Pemilihan tipe <i>Basic</i> I/O CJ1W-ID211.....	29
Gambar 2.26. Pemilihan tipe <i>Basic</i> I/O CJ1W-OC211	29
Gambar 2.27. Bagian-bagian <i>Visual Basic</i> 6.0.....	31
Gambar 2.28. <i>Toolbox</i>	32
Gambar 2.29. Skema Komunikasi <i>MSComm</i>	41
Gambar 2.30. Konfigurasi Kartesian	44
Gambar 2.31. Fotodioda.....	45
Gambar 2.32. Contoh RFID <i>tag</i>	47
Gambar 2.33. RFID <i>Reader</i> ID-12	48
Gambar 2.34. Kaidah Tangan Kiri Flemming.....	49
Gambar 2.35. Mekanisme Kerja Motor DC Magnet Permanen dengan Cincin Belah	49
Gambar 2.36. Dua Roda pada Poros yang Sama.....	51
Gambar 2.37. Dua Roda Dihubungkan Satu Rantai	52
Gambar 2.38. Dua Roda Saling Bersinggungan.....	53
Gambar 2.39. Kontak-kontak <i>Limit Switch</i>	55
Gambar 2.40. Bentuk Fisik Macam-macam <i>Limit Switch</i>	56
Gambar 2.41. Rangkaian Internal Relai.....	56

Gambar 2.42. Macam-macam relai	57
Gambar 2.43. Solenoid Doorlock	58
Gambar 2.44. Diagram Blok Alat.....	60
Gambar 3.1. Penyimpanan Tampak Luar	63
Gambar 3.2. Pemetaan Slot	63
Gambar 3.3. Penyimpanan Tampak Dalam	64
Gambar 3.4. Rangkaian <i>Transmitter</i>	65
Gambar 3.5. Rangkaian <i>Receiver</i>	65
Gambar 3.6. Komunikasi PC, PLC, dan RFID	67
Gambar 3.7. Instalasi RFID.....	68
Gambar 3.8. Instalasi Motor Listrik	68
Gambar 3. 9. <i>Wiring Diagram Prototype</i> Panel Penyimpanan Benda Otomatis ..	69
Gambar 3.10. Flowchart <i>Prototype</i> Penyimpanan Benda Otomatis.....	74
Gambar 4. 1. Pengujian Melalui <i>Hyperterminal</i>	96
Gambar 4. 2. Menu Ambil Pengguna	102

DAFTAR TABEL

	Halaman
Tabel 2.1. Perbandingan Sistem Kontrol	8
Tabel 3.1. Alamat <i>Input Digital</i> PLC	77
Tabel 3.2. Alamat <i>Output Digital</i> PLC	78
Tabel 3.3. Alamat <i>Visual Basic</i> PLC	78
Tabel 3.4. Pengujian Sensor Infra Merah	81
Tabel 3.5. Pengujian RFID.....	82
Tabel 3.6. Pengujian Tegangan pada <i>Prototype</i>	83
Tabel 3.7. Pengujian Sistem Pengaman.....	84
Tabel 3.8. Pengujian Tombol Penyimpanan	85
Tabel 3.9. Pengujian Tombol Pengambilan	85
Tabel 3.10. Pengujian Waktu Tempuh Penyimpanan.....	86
Tabel 3.11. Pengujian Waktu Tempuh Pengambilan	86
Tabel 4.1. Hasil Pengujian Sensor Infra Merah	87
Tabel 4.2. Hasil Pengujian RFID.....	88
Tabel 4.3. Hasil Pengujian Tegangan <i>Prototype</i>	89
Tabel 4.4. Hasil Pengujian Sistem Pengaman.....	90
Tabel 4.5. Hasil Pengujian <i>Penyimpanan Software</i>	91
Tabel 4.6. Hasil Pengujian Pengambilan <i>Software</i>	92
Tabel 4.7. Hasil Pengujian Waktu Kerja Penyimpanan.....	93
Tabel 4.8. Hasil Pengujian Waktu Kerja Pengambilan	93

BAB I

PENDAHULUAN

1.1. Latar Belakang

Menurut Badan Pusat Statistika (2015) di Jabodetabek angka kriminalitas masih tergolong tinggi yaitu sebesar 49.498 kasus pada tahun 2013. Hal ini karena di negara berkembang, termasuk Indonesia, suatu mekatronik sederhana dirasa masih belum menyentuh semua aspek. Tingkat keamanan dan efektifitas yang minim dikarenakan masih banyak hal-hal sederhana yang di negara maju sudah memiliki inovasi teknologi di Indonesia masih menggunakan cara konvensional.

Dalam hal penyimpanan, baik itu benda berharga atau kendaraan bermotor masih banyak kasus seperti keluhan, rusaknya benda yang disimpan, bahkan hilang karena kurang cekatannya staff penyimpanan tersebut. Di samping itu faktor minimnya pengamanan penyimpanan menyebabkan kriminalitas khususnya pencurian belum dapat dikurangi secara besar-besaran.

Hal yang sudah dijelaskan tersebut membuat banyak konsumen khususnya yang berhubungan dengan tempat penyimpanan benda berharga, seperti *safe deposit box* di perbankan, menjadi kecewa. Atau di badan pusat penelitian, di mana alat uji yang digunakan disimpan di tempat yang masih kurang aman, sehingga sering terjadi kesalahan perhitungan karena alat uji mengalami kerusakan pada saat penyimpanan. atau lebih buruknya kemungkinan hilangnya alat uji dan perusahaan harus mengeluarkan biaya tambahan untuk biaya kualitas

atau biaya keluhan. Yang dimaksud biaya kualitas adalah biaya kegagalan pelayanan, pemrosesan keluhan, dan kerugian finansial akibat pelayanan yang mengecewakan.

Oleh karena itu perlu lah sekiranya manajemen penyimpanan khususnya benda dengan mengaplikasikan metode yang lebih modern dibandingkan kunci konvensional seperti pada *safe deposit box* di perbankan. Metode modern seperti *password* atau *Radio Frequency Identification (RFID)*, yang banyak digunakan saat ini, untuk setiap penyimpanan dan pengambilan benda dirasa lebih aman dan sejalan dengan perkembangan jaman. Sehingga perusahaan dapat meminimalisir biaya kualitas atau keluhan.

Dengan mudahnya didapat teknologi modern, seperti *Programmable Logic Controller (PLC)*, dirasa bukan hal yang tidak mungkin untuk menerapkan penyimpanan benda dengan sistem otomasi sehingga meningkatkan efektifitas. Yang dimaksud efektifitas disini adalah pada bagian memilih tempat yang kosong pada rak yang memiliki banyak tempat untuk menyimpan benda. Sehingga tidak diperlukan waktu yang lama untuk berfikir atau memproses benda akan disimpan di mana.

Walaupun menggunakan perangkat otomasi PLC sebagai basis alatnya, perusahaan tetap tidak perlu mengeluarkan biaya yang begitu besar. Biaya tersebut bisa dibandingkan dengan membuat perangkat otomasi menggunakan relai konvensional atau biaya keluhan jika benda yang dititipkan mengalami kerusakan bahkan hilang.

Sehingga dari penjelasan tersebut penulis memutuskan untuk melakukan penelitian eksperimen laboratorium mengenai “*Prototype Penyimpanan Benda Bertingkat Otomatis dengan Pengaman Password dan RFID Berbasis PLC Menggunakan Human Machine Interface.*”

1.2. Identifikasi Masalah

Dari latar belakang yang sudah diuraikan tersebut dapat disimpulkan masalah-masalah yang muncul adalah sebagai berikut :

1. Bagaimanakah cara untuk menutupi sistem pengaman konvensional seperti kunci yang pada jaman sekarang sudah mudah untuk dibobol?
2. Bagaimanakah agar sistem penyimpanan benda bisa bekerja secara otomatis dan bisa diandalkan?
3. Bagaimanakah membuat penyimpanan bertingkat yang semua benda disimpan secara acak oleh pengguna tanpa diketahui pengguna lain?
4. Bagaimanakah rancangan pengaman penyimpanan benda otomatis berdasarkan *password* dan RFID?

1.3. Pembatasan Masalah

Setelah teridentifikasinya masalah, dalam pembuatan *prototype* penyimpanan benda bertingkat otomatis berbasis pada PLC CJ1M ini dibatasi pada pembuatan sistem pengamanan menggunakan *password* dan RFID pada setiap penyimpanan dan pengambilan benda, *interface* antara *prototype* dan pengguna serta

pengendalian perangkat elektromekanik dalam *prototype* penyimpanan benda yang direncanakan memiliki 3 tingkat dan tiap-tiap tingkat memiliki 2 slot.

1.4. Perumusan Masalah

Berdasarkan latar belakang, identifikasi masalah dan pembatasan masalah, peneliti dapat menyimpulkan perumusan masalah yang muncul adalah bagaimanakah membuat *Prototype* Penyimpanan Benda Bertingkat Otomatis dengan Pengaman *Password* dan RFID Berbasis PLC Menggunakan *Human Machine Interface*?

1.5. Tujuan Penelitian

Adapun tujuan dari pembuatan penyimpanan benda bertingkat otomatis berbasis PLC dengan pengaman *password* dan RFID menggunakan *interface software visual basic 6.0* antara lain:

1. Merancang dan membuat penyimpanan benda bertingkat otomatis menggunakan PLC CJ1M.
2. Membuat sistem pengaman menggunakan *password* dan RFID serta menggunakan *visual basic 6.0* sebagai *interface* untuk penyimpanan dan pengambilan benda.

1.6. Manfaat Penelitian

Dari hasil penelitian dan perancangan alat diharapkan memiliki manfaat sebagai berikut :

1. Manfaat untuk peneliti:
 - a. Sebagai usaha untuk memaksimalkan kemampuan dan keterampilan peneliti maupun pembaca dalam menggunakan serta menerapkan PLC dalam kehidupan sehari-hari.
 - b. Salah satu bentuk dalam mengembangkan dan merealisasikan pengetahuan yang diterima selama menempuh pendidikan dan pembelajaran di Universitas Negeri Jakarta.
2. Manfaat akademis:
 - a. Sebagai sumber referensi dalam pembelajaran pengendalian penyimpanan benda bertingkat otomatis berbasis pada PLC CJ1M dengan pengaman *password* dan RFID menggunakan *interface software visual basic 6.0*.
 - b. Sebagai media alat bantu dalam memahami proses data PLC CJ1M dan *visual basic* pada penyimpanan benda bertingkat otomatis dengan pengaman *password* dan RFID.
 - c. Sebagai sumber referensi untuk penelitian dan pengembangan selanjutnya.

3. Kegunaan praktis

- a. Bagi instansi yang memerlukan tempat penyimpanan yang aman, maka akan lebih terjamin keamanannya jika dikendalikan secara otomatis serta berpengaman ganda dan modern seperti *password* dan RFID.

BAB II

KAJIAN TEORITIS DAN KERANGKA BERFIKIR

2.1. *Programmable Logic Controller (PLC)*

2.1.1. Pengertian PLC

Awalnya PLC banyak dikenal sebagai akronim dari PC (*Personal Computer*).¹ Namun dari referensi lain menyatakan bahwa PC yang dimaksud di sini adalah *Programmable Controller*, bukan *Personal Computer*. Karena di beberapa negara pada awalnya terdapat beberapa istilah untuk menjelaskan PLC tersebut. Seperti PC (*Programmable Computer*) yang biasa digunakan oleh orang Inggris, PLC (*Programmable Logic Controller*) yang biasa digunakan oleh orang Amerika, dan PBS (*Programmable Binary System*) yang biasa digunakan oleh orang Swedia.²

Namun pada akhirnya *Programmable Logic Controller* memiliki akronim sendiri agar tidak terjadi kerancuan dengan *Personal Computer*, yaitu PLC. Padahal secara fisik maupun fungsi-nya PLC dan PC memiliki beberapa perbedaan yang sangat menonjol, hal tersebut dapat dilihat pada tabel 2.1.

¹ Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, Universitas Negeri Jakarta, Jakarta, 2012, hlm. 1.

² Ian G. Warnock, *PROGRAMMABLE CONTROLLERS, Operation and Application.*, Prentice Hall International (UK) Ltd., Cambridge, 1988, hlm. 50.

Tapi dengan seiring meningkatnya pemahaman tentang teknologi kini PLC memiliki pengertian sendiri yaitu (*Programmable Logic Controller*).

Tabel 2.1. Perbandingan Sistem Kontrol

KARAKTERISTIK	SISTEM RELAI	DIGITAL LOGIC	PC	PLC
Ukuran fisik	Besar	Sangat kompak	Cukup Besar	Sangat Kompak
Kecepatan Op.	Lamban	Sangat Cepat	Cukup Cepat	Cepat
Harga / Fungsi	Murah	Murah	Mahal	Cukup Mahal
Gangguan <i>Noise</i>	Sangat Baik	Baik	Cukup Baik	Baik
Perubahan Sistem	Sangat Sulit	Sulit	Cukup Mudah	Mudah
Perawatan	Sulit (Banyak Kontak)	Sulit (Komponen disolder)	Sulit (<i>Board</i> dirancang khusus)	Mudah
Pemasangan	Merancang dan memasang sangat lama	Merancang butuh waktu yang lama	Memprogram butuh waktu yang cukup lama	Memprogram dan memasang butuh waktu yang singkat

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)

Programmable Logic Controller menurut *National Electrical Manufactures Assosiation* (NEMA) adalah sebuah perangkat elektronika *digital* yang menggunakan memori yang dapat diprogram dan diprogram ulang sebagai penyimpanan internal dan menyediakan instruksi-instruksi untuk menjalankan fungsi-fungsi yang spesifik seperti *Logic, Sequence, Timing, Counting* dan *Arithmetic*.³

Seiring perkembangan teknologi *solid state*, saat ini PLC telah mengalami perkembangan luar biasa, baik dari ukuran, kepadatan komponen serta dari

³ Syufrijal, *Op.cit.*, hlm. 3.

segi fungsionalnya. Beberapa peningkatan perangkat keras dan perangkat lunak ini diantaranya adalah:

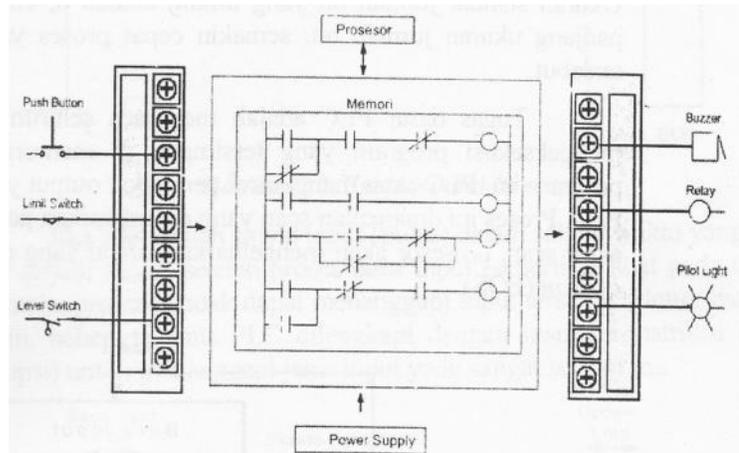
- 1) Ukuran semakin kecil dan kompak
- 2) Jumlah *input/output* yang semakin banyak dan padat.
- 3) Beberapa jenis dan tipe PLC dilengkapi dengan modul-modul untuk tujuan kontrol kontinyu, misalnya modul ADC/DAC, PID, modul *Fuzzy*, dan lain-lain.
- 4) Pemrograman relatif semakin mudah. Hal ini terkait dengan perangkat lunak pemrograman yang *user friendly*.
- 5) Memiliki kemampuan komunikasi dan sistem dokumentasi yang semakin baik.
- 6) Jenis instruksi/fungsi semakin banyak dan lengkap.
- 7) Waktu eksekusi program semakin cepat.⁴

2.1.2. Komponen Utama PLC

Penjelasan mengenai komponen utama PLC memiliki beberapa perbedaan dalam beberapa literatur. Ada yang menyatakan bahwa perangkat keras PLC pada dasarnya tersusun dari empat komponen utama berikut: *processor, power*

⁴ Iwan Setiawan, *Programmable Logic Controller (PLC) dan Teknik Perancangan Sistem Kontrol*, Andi, Yogyakarta, 2006, hlm. 2.

supply, memori, dan modul *input/output*. Secara fungsional, interaksi antara keempat komponen penyusun PLC ini dapat diilustrasikan pada gambar 2.1.⁵



Gambar 2.1. Interaksi Komponen-komponen sistem PLC

Sumber : Setiawan, Iwan, *Programmable Logic Controller (PLC) dan Teknik Perancangan Sistem Kontrol*, (Andi : 2006)

Sedangkan dari literatur lain dinyatakan bahwa komponen utama PLC pada prinsipnya tidak jauh berbeda dari perangkat keras yang dimiliki oleh komputer, yaitu terdiri dari *central processing unit* atau CPU, modul *input*, modul *output*, alat pemrograman, dan catu daya.⁶ Ada pula yang menyatakan bahwa PLC terdiri atas lima komponen utama, yaitu *central processing unit* (CPU), *programmer/monitor* (PM), modul *input/output* (I/O), *Printer*, *program recorder/player*.⁷ Maka secara garis besar ditarik kesimpulan bahwa komponen utama PLC terdiri atas : CPU, yang di dalamnya sudah mencakup

⁵ *Idem*, hlm. 51

⁶ Syufrijal, *Op.cit.*, hlm.6

⁷ John W. Webb, *Programmable Logic Controllers Principles and Applications*, Macmillan Publishing Company, New York, 1992, hlm. 10.

processor, dan memori; Modul *Input/Output*, baik itu modul *input/output analog* ataupun *digital*; dan catu daya. Sehingga perangkat pemrograman PLC termasuk ke dalam komponen pelengkap.

2.1.2.1. *Central Processing Unit (CPU)*

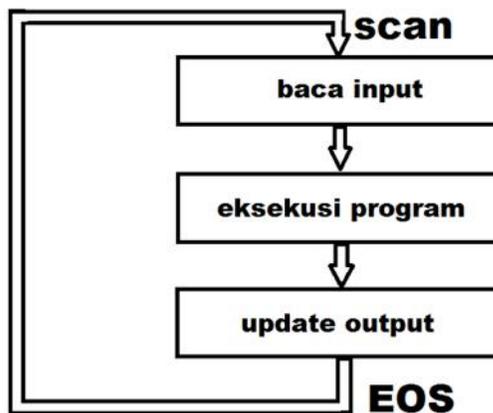
CPU adalah sistem yang didasarkan mikroprosesor yang mengganti relai pengendali, pencacah, *timer*, dan pembuat urutan.⁸ Fungsi CPU untuk memproses kegiatan yang ada pada PLC dengan menjalankan instruksi-instruksi yang telah diprogram dan disimpan dalam memori. Pada CPU PLC memori dibagi menjadi 2 jenis, yaitu RAM (*Random Access Memory*) dan ROM (*Read Only Memory*).

Yang membedakan kedua memori tersebut adalah kemampuan menyimpan program dan fasilitas untuk dilakukan perubahan ulang. Dari kedua jenis tersebut ROM memiliki kemampuan untuk menyimpan data lebih baik, karena ROM kebal terhadap *noise* listrik maupun kehilangan sumber daya. sehingga walaupun PLC dalam keadaan mati, data-data tetap tersimpan di dalam ROM. Namun dalam fasilitas untuk melakukan perubahan ulang RAM jauh lebih unggul, karena RAM dirancang agar informasi data dapat dimasukkan ke dalam memori dan dapat dipanggil setiap saat. Proses perubahan ulang data pada RAM menjadi jauh lebih mudah. Namun pada RAM masih memiliki kekurangan karena masih

⁸ Frank D. Petruzella, *Elektronika Industri. Terj*, Andi, Yogyakarta, 2001, hlm. 598

belum begitu kebal terhadap *noise* listrik, sehingga dimungkinkan data dapat hilang ketika terjadi ketidak-stabilan listrik.

Pada intinya fungsi CPU secara keseluruhan adalah untuk mengatur tugas pada keseluruhan sistem PLC, termasuk di dalamnya fungsi matematis, manipulasi data, diagnostik, dan lain sebagainya. Jelasnya sistem kerja CPU ditunjukkan pada gambar 2.2.



Gambar 2.2. Representasi scan PLC

Sumber : Setiawan, Iwan, *Programmable Logic Controller (PLC) dan Teknik Perancangan Sistem Kontrol*, (Yogyakarta, Andi : 2006)

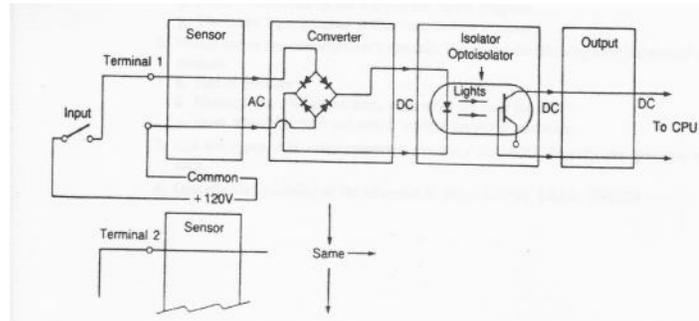
Proses yang diilustrasikan oleh gambar 2.2 dinamakan sebagai *scan*, di mana pada setiap akhir *scan*, CPU akan mengeluarkan sinyal yang dinamakan sinyal *end-of-scan* (EOS). Untuk *processor* yang terdapat dalam PLC dapat dikategorikan berdasarkan panjang atau ukuran jumlah bit dari *register-register processor* tersebut. Sehingga semakin banyak jumlah bit yang tersedia, semakin cepat pula proses *scan* yang terjadi pada PLC tersebut.

Waktu *scan* ini secara umum dipengaruhi oleh 2 faktor utama: (1) jumlah memori yang diperlukan oleh program PLC (jumlah anak tangga pada diagram ladder) dan (2) jenis instruksi yang digunakan dalam program.⁹ Kendati demikian waktu scan PLC berkisar antara beberapa mili detik hingga ratusan mili detik.

2.1.2.2. Modul *Input* dan *Output* (I/O)

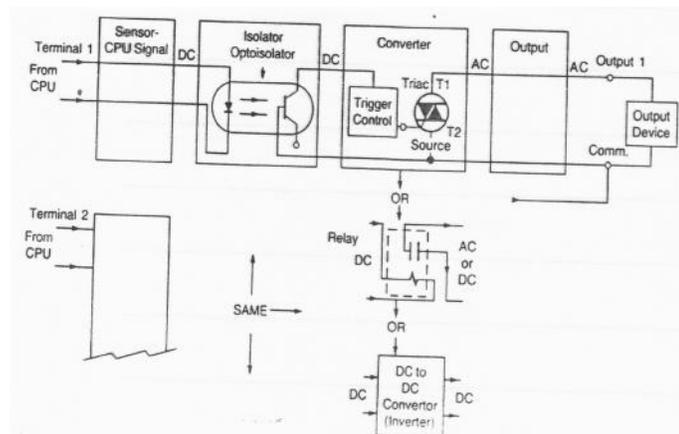
Modul *input* dan *output* (I/O) merupakan sebuah rangkaian perangkat antarmuka, yang merupakan penghubung antara CPU dengan alat I/O. *Layout* hubungan dari modul I/O dengan CPU ditunjukkan oleh gambar 2.3 dan gambar 2.4. Fungsinya untuk mengubah sinyal dari alat I/O menjadi sinyal yang dimengerti oleh PLC baik itu sinyal *analog* maupun *digital*. Untuk kontrol yang besar, dikenal modul *remote* I/O. Di mana modul ini bisa bekerja pada jarak yang cukup jauh yaitu ± 10000 *feet*, sehingga akan menghemat biaya untuk instalasinya. Sesuai dengan namanya modul *input* merupakan komponen PLC yang menerima sinyal masukan dari proses luar. Sedangkan modul *output* merupakan komponen PLC yang memberikan sinyal akhir ke proses luar PLC.

⁹ Iwan Setiawan, *Op.cit*, hlm.53



Gambar 2.3. Layout Modul Input

Sumber : John W. Webb, *Programmable Logic Controllers Principles and Applications.*, (New York, Macmillan Publishing Company : 1992)



Gambar 2.4. Layout modul Output

Sumber : John W. Webb, *Programmable Logic Controllers Principles and Applications.*, (New York, Macmillan Publishing Company : 1992)

Pada PLC ukuran sedang atau kecil biasanya modul I/O sudah menyatu dengan CPU dan *power supply*. Modul I/O mempunyai kapasitas titik sambungan tiap *channel* yang bergantung pada jumlah *word* dari PLC tersebut. Sehingga jenis PLC mempengaruhi pada jumlah I/O yang bisa digunakan.

2.1.2.3. *Power Supply*

Umumnya *power supply* PLC membutuhkan tegangan masukan dari sumber AC yang besarnya bervariasi antara 120 hingga 220 VAC dengan toleransi masukan antara 10-15%. Hanya sebagian kecil PLC yang membutuhkan tegangan *input* langsung dari sumber DC (umumnya, besar sumber tegangan ini adalah 24 VDC). Unit *power supply* pada PLC-PLC ukuran kecil umumnya sudah menyatu dengan PLC-nya, sedangkan ada juga unit *power supply* yang terpisah dari modul lainnya.

Jika batas toleransi tegangan tidak terpenuhi pada umumnya komponen *power supply* akan mengirim perintah ke CPU untuk mematikan sistem PLC tersebut. Untuk kasus-kasus tertentu di mana arus kurang memenuhi kebutuhan dari CPU namun tetap dipaksakan mengeksekusi program, maka akan terjadi situasi *undercurrent* yang sering menyebabkan kesalahan *interminant*.¹⁰ Kesalahan *interminant* ini adalah kesalahan yang ketika terjadi akan sulit untuk diketahui dari mana penyebabnya.

2.1.2.4. Perangkat Pemrograman

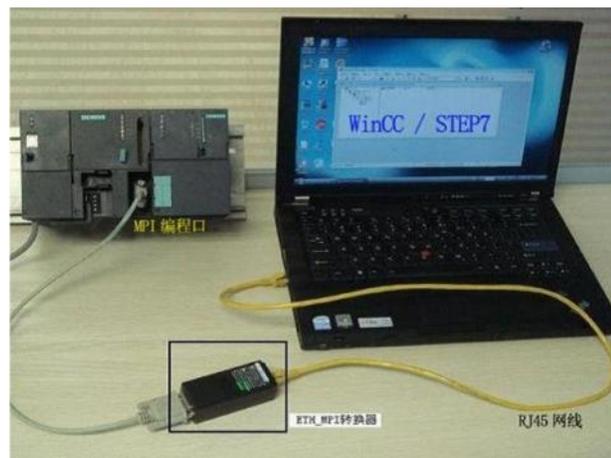
Untuk memberikan perintah atau program pada PLC terdapat dua cara, yaitu menggunakan *Miniprogrammer/console* atau menggunakan *Personal Computer* (PC). Yang membedakan dari kedua cara tersebut terletak pada mekanisme untuk memasukkan program ke dalam memori PLC tersebut.

¹⁰ *Ibid.*, hlm. 54.

langsung di lapangan. Sedangkan untuk memasukkan program secara keseluruhan pada PLC biasanya menggunakan PC. Karena jika jumlah anak tangga pada *ladder diagram* berukuran relatif besar akan melelahkan pengguna *console*.

2.1.2.4.2. Personal Computer (PC)

Berkaitan dengan arsitekturnya yang bersifat *general purpose* dan sistem operasinya yang standar, umumnya vendor-vendor PLC menyertakan perangkat lunak PC untuk mengimplementasikan pemasukan program, peng-*edit*-an, dokumentasi dan program *monitoring real time* pada PLC. Seperti program *cx-programmer* pada PLC keluaran OMRON. Bahkan banyak perangkat lunak tersebut yang sudah dilengkapi simulasi dengan simbol-simbol perangkat masukan dan keluaran secara visual. Contoh visualisasi konektifitas PC ke PLC ditunjukkan pada gambar 2.6.



Gambar 2.6. Konektifitas PC ke PLC

Sumber : <http://nl.aliexpress.com/w/wholesale-omron-cable.html>

Dengan adanya perangkat lunak tersebut menjadikan pengguna lebih mudah dalam memasukkan program. Setiap program langsung divisualisasikan, baik itu *ladder diagram* atau kode *mnemonic*, tidak seperti pada *console* yang hanya berupa susunan kode-kode *mnemonic*.

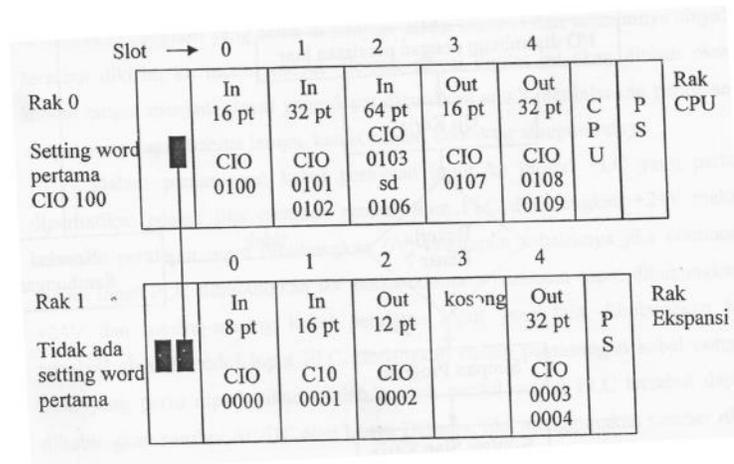
2.1.3. Pengalamatan I/O PLC

Modul I/O yang digunakan sebagai *input* atau *output* pada PLC jenis CS1 dan CJ1 tidak akan mempengaruhi alokasi I/O *word* pada unit tersebut. Alokasi I/O ditentukan berdasarkan pemasangan slot pada rak CPU. Penomoran slot pada modul I/O dimulai dari kiri. Alokasi I/O dasar pada rak CPU CS1 atau CJ1 dalam bentuk *word* dapat dilihat pada gambar 2.7 sedangkan pengalamatan I/O pada rak ekspansi CPU dapat dilihat pada gambar 2.8.

		Slot → 1	2	3	4	5	
P S	CPU	In 16 pt	In 16 pt	In 32 pt	Out 32 pt	Out 64 pt	Rak CPU
		0000	0001	0002 0003	0004 0005	0006 sd 0009	

Gambar 2.7. Pengalamatan I/O CS1/CJ1

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)



Gambar 2.8. Pengalamatan I/O CS1/CJ1 pada Rak Ekspansi

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)

Pada PLC CS1 dapat menggunakan rak ekspansi CPU hingga 7 buah dengan maksimal 10 unit I/O pada setiap rak sedangkan pada PLC CJ1 hanya dapat menggunakan 3 rak ekspansi CPU dengan maksimal 10 unit I/O pada setiap rak. Seperti yang terlihat pada gambar 2.8, pengalamatan I/O pada rak 0 *word* diatur pada CIO 100 sedangkan pada rak 1 tidak ada pengaturan *word* pertama sehingga pengalamatan I/O dimulai dari CIO 000. Pengaturan *word* pertama dapat dilakukan menggunakan program *CX-Programmer*. Adapun langkah-langkah yang harus dilakukan adalah pilih *Rack Start Adress* pada menu *option I/O table*, kemudian hilangkan tanda centang dalam tulisan *invalid* pada *main rak*.

2.1.4. Instruksi Dalam Pemrograman PLC

2.1.4.1. Instruksi-instruksi Dasar

Instruksi dasar merupakan instruksi yang digunakan untuk membuat rangkaian logika dari diagram tangga. Instruksi dasar ini ada enam, yaitu : *LD*, *NOT*, *OUT*, *AND*, *OR*, dan *END*. Fungsi dari instruksi-instruksi dasar tersebut adalah sebagai berikut:

1) *LD*

LD atau singkatan dari *Load*, yang merupakan instruksi untuk memulai program garis atau blok pada rangkaian logika yang dimulai dengan kontak *normally open* (NO).

2) *NOT*

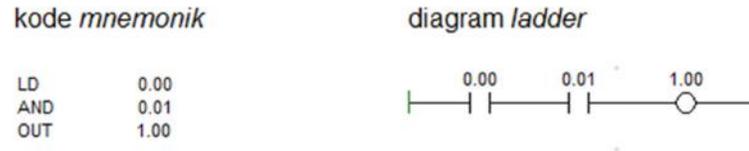
Instruksi dasar *NOT* berfungsi untuk membentuk suatu kontak *normally close* (NC).

3) *OUT*

OUT merupakan instruksi untuk memasukkan koil *output*. Kontak-kontak dari masing-masing koil *output* dapat digunakan beberapa kali sesuai dengan yang diinginkan.

4) *AND*

Instruksi *AND* digunakan untuk menghubungkan dua atau lebih kontak-kontak *input output* secara seri. Instruksi *AND* dapat dilihat pada gambar 2.9.

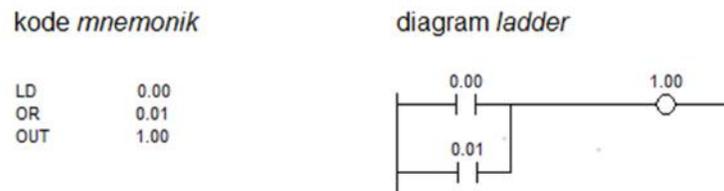


Gambar 2.9. Instruksi AND

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)

5) *OR*

Instruksi dasar *OR* digunakan untuk menghubungkan dua atau lebih kontak-kontak *input* atau *output* secara paralel. Instruksi *OR* dapat dilihat pada gambar 2.10.



Gambar 2.10. Instruksi OR

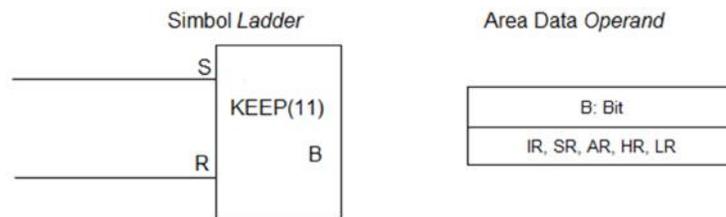
Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)

6) *END*

Instruksi dasar *END* untuk menyatakan rangkaian kontrol yang dibuat telah berakhir. Instruksi *END* ini harus selalu dimasukkan dalam penulisan program, karena apabila akhir rangkaian kontrol tidak dilengkapi dengan instruksi *END*, maka program tersebut tidak akan dieksekusi oleh CPU.

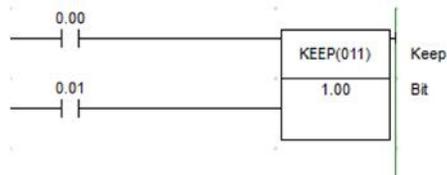
2.1.4.2. Instruksi *KEEP*

Instruksi *KEEP* digunakan untuk memaksa hasil keluaran menjadi *ON*. Pada PLC Omron instruksi *KEEP* ini sama fungsinya dengan instruksi *SET/RESET*. Bedanya pada penulisan program instruksi *KEEP* sinyal *input* untuk *SET* dan *RESET*-nya digabungkan menjadi satu blok. Simbol *ladder* instruksi ini dapat dilihat pada gambar 2.11 sedangkan contoh penggunaan instruksi *KEEP* dapat dilihat pada gambar 2.12.



Gambar 2.11. Simbol Ladder dan Area Data Operand instruksi Keep

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)



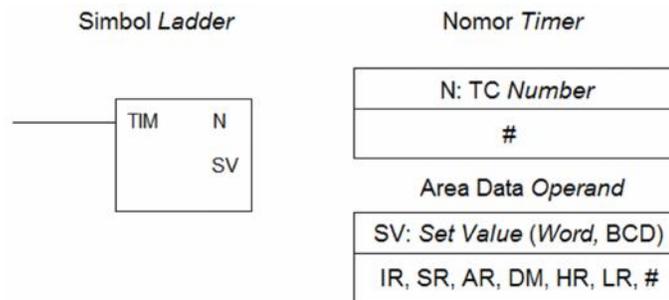
Gambar 2.12. Contoh Penggunaan Instruksi Keep

Sumber : Dokumen pribadi

2.1.4.4. Instruksi *Timer*

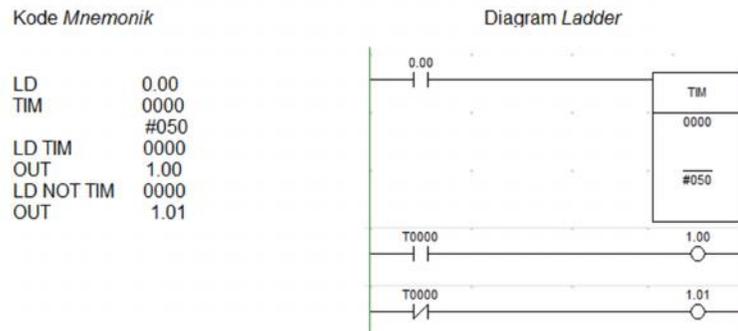
Timer berfungsi untuk mengaktifkan suatu keluaran dengan interval waktu yang dapat diatur. Pengaturan waktu dilakukan melalui nilai *setting* (*preset value*). *Timer* pada PLC Omron diberi nomor dari 0-127 (T0 – T127). Instruksi *timer* ada 2 macam yaitu *Timer* (TIM) dan *High Timer* (TIMH). Perintah TIM dan TIMH pada dasarnya sama, yaitu berfungsi

sebagai *timer*. Bedanya pada pengukuran waktu TIM mempunyai pulsa *clock* lebih panjang dibandingkan TIMH. TIM mempunyai pulsa *clock* sebesar 0,1 detik sedangkan TIMH mempunyai pulsa *clock* sebesar 0,01 detik. *Timer* tersebut akan bekerja bila diberi *input* dan mendapat pulsa *clock*. Untuk pulsa *clock* sudah disediakan oleh pembuat PLC. Simbol *ladder timer* dan contoh penggunaan *timer* dapat dilihat pada gambar 2.13 dan 2.14.



Gambar 2.13. Simbol Ladder, Nomor, dan Area Data Operand Timer

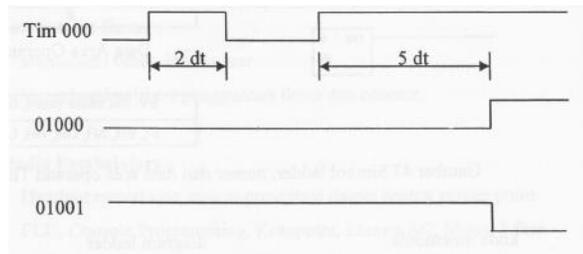
Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)



Gambar 2.14. Contoh Penggunaan Timer

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)

Cara kerja *timer* pada gambar 2.14 juga dapat dilihat pada gambar 2.15.

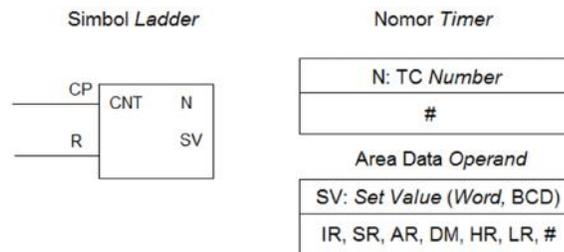


Gambar 2.15. Diagram Waktu *Timer*

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)

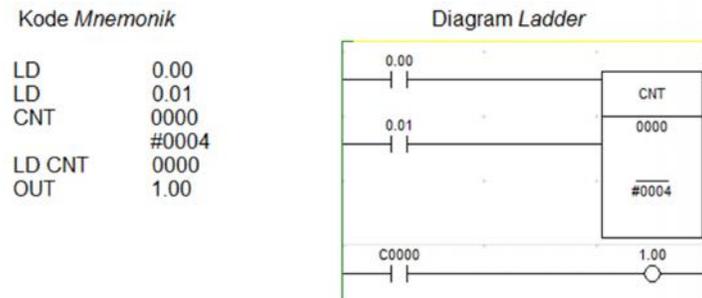
2.1.4.5. Instruksi *Counter*

Pada PLC Omron terdapat *counter* yang diberi nomor dari 0 – 127 (C0-C127). Penggunaan alamat *counter* ini digunakan secara bersama-sama dengan *timer*. Oleh sebab itu dalam satu program pemberian nomor *counter* tidak boleh sama dengan nomor *timer*. Cara kerja *counter* dan *timer* mirip, perbedaannya *timer* mencacah pulsa internal sedangkan *counter* mencacah pulsa dari luar. Ada 2 sinyal *input* yang digunakan oleh *counter* yaitu sinyal pulsa dan sinyal *reset*. Simbol *ladder counter* dalam program diagram *ladder* dan kode *mnemonik* dapat dilihat pada gambar 2.16 sedangkan contoh penggunaan *counter* dapat dilihat pada gambar 2.17.



Gambar 2.16. Simbol *Ladder*, Nomor dan Area Data Operand *Counter*

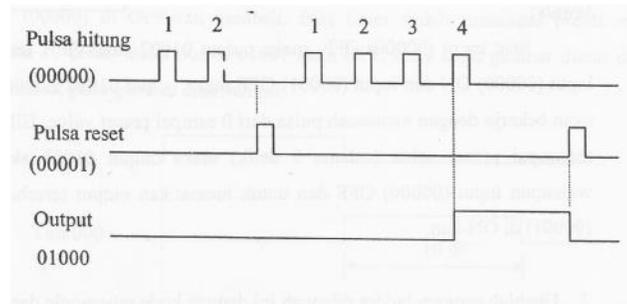
Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)



Gambar 2.17. Contoh Penggunaan Counter

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)

Cara kerja counter pada gambar 2.17 dapat juga dilihat pada gambar 2.18.

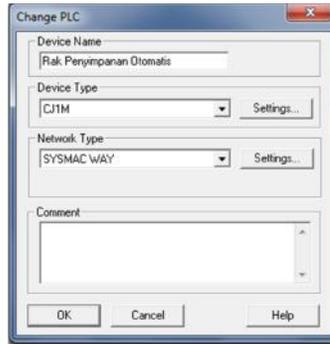


Gambar 2.18. Diagram Waktu Counter

Sumber : Syufrijal, *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC.*, (Universitas Negeri Jakarta : 2012)

2.1.5. Pengaturan Hardware PLC

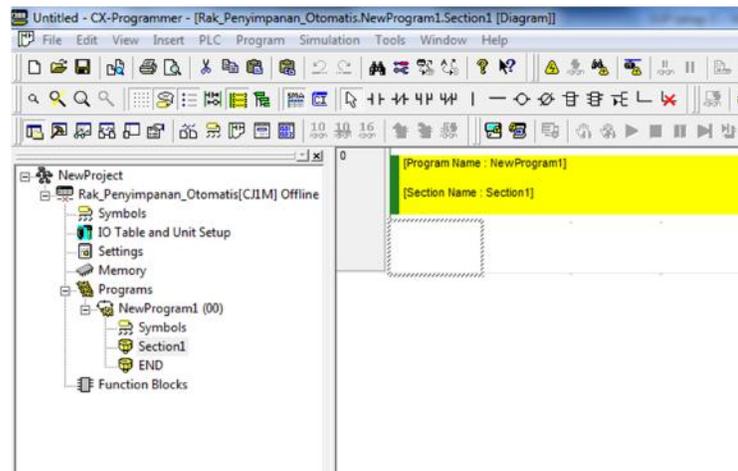
Perancangan *hardware* PLC yang akan diproses oleh PLC, pertama yang harus dilakukan adalah mengkonfigurasi penggunaan *hardware* pada program yang dibuat. Konfigurasi harus tepat sama dengan PLC yang akan digunakan. Karena program hanya akan dapat dikirimkan dan dieksekusi apabila penentuan *hardware* dalam pemrograman sesuai dengan PLC yang digunakan. Konfigurasi *hardware* PLC Omron dilakukan dengan menggunakan *software* CX-Programmer. *Hardware* PLC yang digunakan untuk mengendalikan



Gambar 2.21. Setting awal PLC

Sumber : Dokumen pribadi

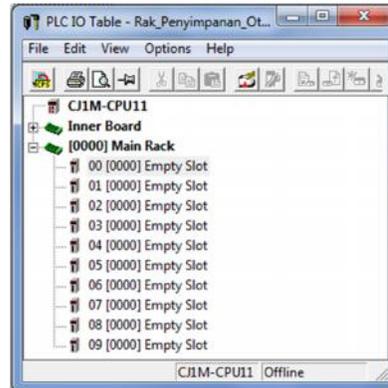
3. Pada jendela *new* seperti pada gambar 2.21 isi kolom *device name* dengan judul program “rak penyimpanan otomatis”. Lalu pada kolom *device type* pilih jenis PLC CJ1M dan kemudian klik *setting*. Setelah itu pilih *CPU type* dan pilih CPU11 lalu OK. Kemudian pada kolom *network type* pilih *sysmacWay* lalu OK, sehingga akan muncul tampilan seperti gambar 2.22.



Gambar 2.22. Tampilan Awal Program

Sumber : Dokumen pribadi

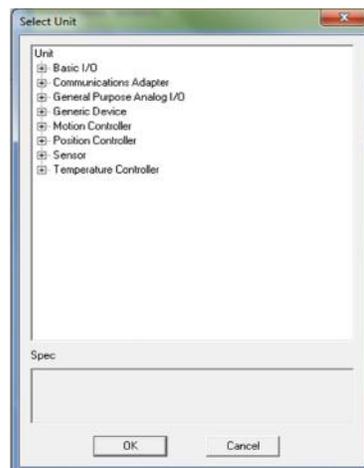
4. Pada gambar 2.22 klik dua kali *IO table and Unit Setup* untuk mengkonfigurasi *hardware*. Maka akan muncul tampilan gambar 2.23.



Gambar 2.23. Tampilan Konfigurasi Hardware

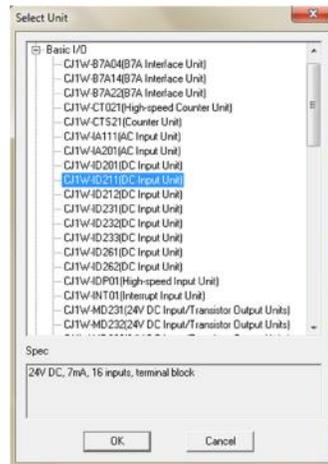
Sumber : Dokumen pribadi

5. Selanjutnya pada gambar 2.23 terlihat pada *main rack* semua *slot* yang belum terisi. Karena pada PLC yang terpasang ada 2 modul maka *slot* 00 hingga *slot* 01 kita isi sesuai modul yang terpasang. Untuk mengkonfigurasi *slot* 00, klik dua kali pada wilayah kosong pada slot 00 sehingga muncul pilihan seperti pada gambar 2.24. Pilihan modul harus disesuaikan dengan PLC yang digunakan. Untuk *slot* 00, modul yang terpasang adalah modul *basic I/O tipe* CJ1W-ID211. Pemilihan modul CJ1W-ID211 seperti terlihat pada gambar 2.25.



Gambar 2.24. Pilihan Hardware Unit PLC

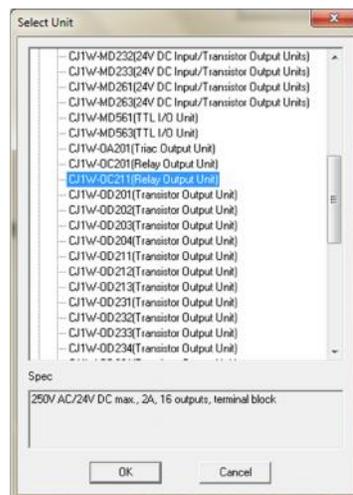
Sumber : Dokumen pribadi



Gambar 2.25. Pemilihan tipe Basic I/O CJ1W-ID211

Sumber : Dokumen Pribadi

6. Slot berikutnya yaitu slot 01. Modul yang terpasang adalah modul *Basic I/O* tipe CJ1W-OC211. Tekan pada modul yang dimaksud lalu tekan OK seperti pada gambar 2.26 dan pengaturan rack pada PLC CJ1M alat ini telah selesai dilakukan.



Gambar 2.26. Pemilihan tipe Basic I/O CJ1W-OC211

Sumber : Dokumen Pribadi

2.2. Visual Basic

Visual basic (atau sering disingkat VB) adalah perangkat lunak untuk menyusun program aplikasi yang bekerja dalam lingkungan sistem operasi *windows*. Dengan *visual basic* kita bisa memanfaatkan kemampuan *windows* secara optimal. Dengan kecanggihannya yang ditawarkan oleh *visual basic* kita akan merasakan begitu mudahnya menyusun program aplikasi dengan tampilan grafis yang menawan dalam waktu yang relatif singkat.¹¹

2.2.1. Kelebihan Visual Basic

Sebagai program berbasis *desktop visual basic* memiliki beberapa kelebihan, yaitu :

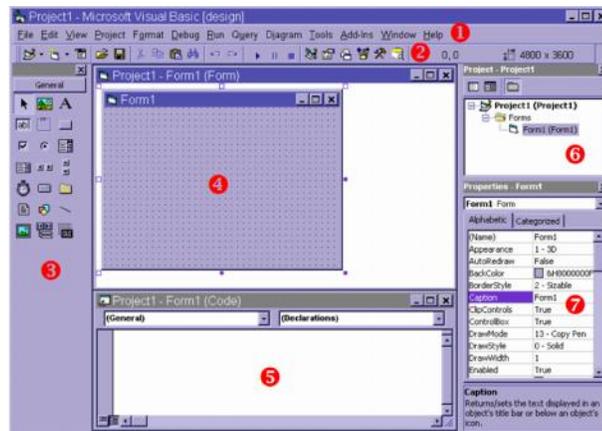
1. *Visual basic* merupakan *high-level programming*. Yang disebut *high-level programming* adalah pemrograman yang telah menggunakan *script-script* “bahasa manusia”, sehingga lebih mudah dimengerti, terutama oleh orang awam.
2. *Visual basic* merupakan aplikasi yang memiliki fitur IDE (*Integrated Development Environment*). Itu artinya, di dalam aplikasi *visual basic* sendiri, kita akan menemukan banyak fitur siap pakai (misalnya *toolbox* untuk membuat tombol atau *textbox*) yang bisa langsung diintegrasikan dengan *script-script* pemrograman.

¹¹ Retna Prasetia, *Teori dan Praktek Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0*, Andi, Yogyakarta, 2004, hlm. 3.

3. *Visual basic* adalah bahasa pemrograman yang bersifat *event-driven*. Jika dibahasakan dalam istilah sehari-hari, *event-driven programming* merupakan *script* pemrograman yang “bereaksi” apabila ada kejadian (*event*) yang dipicu oleh seorang *user* misalnya saat menekan tombol “*Submit*”, “*Hitung*” dan sebagainya.
4. Secara umum, ekosistem *visual basic* sudah terbentuk. Kita bisa menemukan referensi yang melimpah, contoh *script* yang banyak ditemukan, dan aplikasi-aplikasi pendukung yang bisa dipasang secara terpisah untuk “memperkuat” daya mampu *visual basic* ini.¹²

2.2.2. Bagian-bagian *Visual Basic (Integrated Development Environment)*

Untuk dapat bekerja dengan baik pada *visual basic* tentunya kita harus mengetahui terlebih dahulu bagian-bagian dari jendela kerja atau IDE (*Integrated Development Environment*), seperti ditunjukkan pada gambar 2.27.



Gambar 2.27. Bagian-bagian *Visual Basic 6.0*

Sumber : <https://electricse.wordpress.com/2011/11/09/mengenal-bagian-bagian-utama-pada-ide-vb-6-0/>

¹² Jubilee Enterprise, *Pemrograman Visual Basic 6, Cara cepat bagi Pemula Menguasai VB 6, PT. Elex Media Komputindo, Jakarta, 2015, hlm.2.*

1. *Menubar*

Digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dan sebagainya.

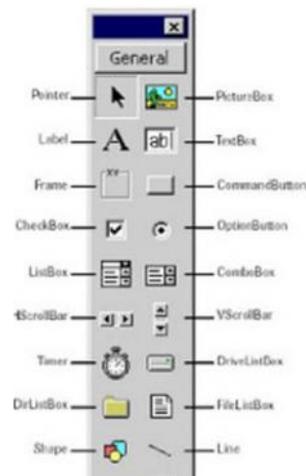
2. *Toolbar*

Digunakan untuk melakukan tugas-tugas tertentu dengan cepat.

3. *Toolbox*

Bila *Toolbox* tidak muncul klik tombol *Toolbox* pada bagian *Toolbar* atau klik menu *View > Toolbox*.

Didalam *toolbox* terdapat beberapa *tool* yang biasa digunakan di dalam *visual basic* seperti ditunjukkan pada gambar 2.28.



Gambar 2.28. Toolbox

Sumber : <https://hinggakujenuh.wordpress.com/belajar-delphi-7/visualbasic/tool-yang-ada-didalam-vb-6-0/>

Adapun nama bagian dan fungsi yang ditunjukkan pada gambar 2.20 yaitu sebagai berikut:

a. *Pointer*

Pointer bukan merupakan suatu kontrol, fungsi ini berguna ketika anda ingin memilih kontrol yang sudah berada pada *form*.

b. *PictureBox*

Kontrol yang digunakan untuk menampilkan gambar dengan format: BMP, DIB (*bitmap*), ICO (*icon*), CUR (*cursor*), WMF (*metafile*), EMF (*enhanced metafile*), GIF, dan JPEG.

c. *Label*

Kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.

d. *TextBox*

Kontrol yang mengandung *string* yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal, atau banyak baris.

e. *Frame*

Kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.

f. *CommandButton*

Merupakan kontrol yang hampir ditemukan pada setiap *form*, dan digunakan untuk membangkitkan *event* proses tertentu ketika pemakai melakukan klik padanya.

g. *CheckBox*

Digunakan untuk pilihan yang isinya bernilai ya atau tidak, benar atau salah.

h. *OptionButton*

Sering digunakan lebih dari satu sebagai pilihan terhadap beberapa pilihan yang hanya dapat dipilih satu.

i. *ListBox*

Mengandung sejumlah *item*, dan *user* dapat memilih lebih dari satu (bergantung pada *property MultiSelect*).

j. *ComboBox*

Merupakan kombinasi dari *TextBox* dan suatu *ListBox* dimana memasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.

k. *HScrollBar* dan *VScrollBar*

Digunakan untuk membentuk *scrollbar* berdiri sendiri.

l. *Timer*

Digunakan untuk proses *background* yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.

m. *DriveListBox*, *DirListBox*, dan *FileListBox*

Sering digunakan untuk membentuk dialog *box* yang berkaitan dengan *file*.

n. *Shape* dan *Line*

Digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.

o. *Image*

Berfungsi menyerupai *image box*, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa kontrol *image* menggunakan sumber yang lebih kecil dibandingkan dengan *PictureBox*.

p. Data

Digunakan untuk data *binding* OLE dapat digunakan sebagai tempat bagi program eksternal seperti Microsoft Excel, Word, dan lain-lain.

4. Jendela *Form*

Tempat munculnya rancangan tampilan program. Bila Jendela *Form* tidak muncul klik tombol *View Object* pada bagian *Project Explorer* atau klik menu *View > Object*.

5. Jendela *Code*

Tempat anda untuk menulis *coding*. Bila Jendela *Code* tidak muncul klik tombol *View Code* pada bagian *Project Explorer* atau klik menu *View > Code*.

6. *Project Explorer*

Bila *Project Explorer* tidak muncul klik tombol *Project Explorer* pada bagian *Toolbar* atau klik menu *View > Project Explorer*.

7. Jendela *Properties*

Merupakan daftar properti-properti objek yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (*foreground*) dan warna latarbelakang (*background*). Anda dapat menggunakan F4 untuk menampilkan jendela *property*. Bila Jendela Properti tidak muncul klik tombol *Properties Window* pada bagian *Toolbar* atau klik menu *View > Properties Window*.¹³

¹³Putra, *tool yang ada di dalam vb 6.0*, <https://hinggakujenuh.wordpress.com/belajar-delphi-7/visualbasic/tool-yang-ada-didalam-vb-6-0/>, diunduh pada 3 Juli 2015.

2.2.3. Kontrol Program *Visual Basic*

2.2.3.1. *IfThen*

Dengan pernyataan ini kita dapat menguji satu kondisi tertentu dan kemudian menentukan suatu tindakan jika kondisi tersebut terpenuhi.

Sintak penulisannya adalah sebagai berikut:

```
If <syarat kondisi> Then <pernyataan>
```

Bisa juga, jika menggunakan *multiple-line*, sebagai berikut:

```
If <syarat kondisi> Then
```

```
  <pernyataan pertama>
```

```
  <pernyataan kedua>
```

```
  -
```

```
  -
```

```
End If
```

Berikut adalah contoh penggunaannya.

```
If Angka = 0 Then
```

```
  Label1.Caption = "Ini adalah angka 0"
```

```
  Label2.Caption = "ini adalah angka 0"
```

```
End If
```

Pernyataan program di atas akan mendeteksi nilai dari variabel Angka. Jika nilainya nol, maka properti *caption* Label1 dan Label2 akan diisi tulisan "ini adalah angka 0". Jika nilainya tidak nol, maka pernyataan tersebut akan diabaikan.

2.2.3.2. *If...Then...Else*

Pernyataan ini hampir sama dengan pernyataan *If...Then*, yaitu digunakan untuk menguji suatu kondisi tertentu. Hanya saja, jika suatu kondisi tidak terpenuhi, maka alur program akan mengeksekusi pernyataan yang lain. Berikut ini adalah sintak penggunaannya.

```

If <syarat kondisi 1> Then
    <blok pernyataan pertama>
ElseIf <syarat kondisi 2> Then
    <blok pernyataan kedua>
    -
    -
ElseIf <syarat kondisi ke-n> Then
    <blok pernyataan ke-n>
Else
    <blok pernyataan>
End If

```

Visual basic pertama kali akan menguji syarat kondisi 1. Jika kondisi 1 tidak dipenuhi, *visual basic* akan menguji kondisi 2 dan seterusnya sampai menemukan kondisi yang terpenuhi. Jika *visual basic* menemukan kondisi yang memenuhi, maka *visual basic* akan mengerjakan blok pernyataannya. Jika *visual basic* tidak menemukan kondisi yang memenuhi, maka blok pernyataan *else* yang akan dieksekusi. Berikut ini adalah contoh penggunaannya.

If Angka = 0 **Then**

Label1.Caption = “Ini adalah angka 0”

ElseIf Angka = 1 **Then**

Label1.Caption = “Ini adalah angka 1”

Else

Label1.Caption = “Ini bukan angka 0 maupun angka 1”

End If

Visual basic akan menguji apakah angka adalah “0”. Jika benar, maka properti *caption* dari Label1 akan diisi “Ini adalah angka 0”. Jika salah, maka *visual basic* akan menguji apakah angka adalah angka “1”. Jika benar, maka properti *caption* dari Label1 akan diisi “Ini adalah angka 1”. Jika salah, maka properti *caption* dari Label1 akan diisi “Ini bukan angka 0 maupun angka 1”.

2.2.3.3. *Select...Case*

Pada dasarnya perintah ini sama dengan perintah *If...Then...Else*, yaitu akan mengeksekusi satu blok pernyataan dari beberapa pilihan blok pernyataan. Hanya saja penulisannya lebih ringkas dan lebih mudah dimengerti. Sintak penulisannya adalah sebagai berikut.

Select Case <kondisi yang diuji>

Case <syarat kondisi 1>

<blok pernyataan pertama>

Case <syarat kondisi 2>

<blok pernyataan kedua>

-

-

Case Else

<blok pernyataan ke-n>

End Select

Berikut adalah contoh penggunaannya.

Select Case Angka***Case 0***

Label11.Caption = "Ini adalah angka 0"

Case 1

Label11.Caption = "Ini adalah angka 1"

Case Else

Label11.Caption = "Ini bukan angka 0 maupun angka 1"

End Select

Eksekusi kode program di atas adalah sama dengan eksekusi *If...Then...Else*, yaitu akan menguji nilai Angka apakah Angka adalah "0", "1", atau lain dari "0" dan "1".¹⁴

2.2.4. Antarmuka *Intelligent Electronic Device* (IED) melalui *Visual Basic*

6.0

Visual basic sudah cukup mumpuni untuk digunakan sebagai sebuah antarmuka antara komputer dengan perangkat *intelligent electronic device* (IED) seperti PLC, RFID, dan lain sebagainya. Untuk dapat berkomunikasi dengan IED, *visual basic* sudah dilengkapi dengan fasilitas *Microsoft Comm*

¹⁴ Retna Prasetia, Op.Cit. hlm 28-31.

Control 6.0. *MS Comm Control* ini berfungsi untuk mengolah jalannya komunikasi dan transfer data dari komputer ke IED atau sebaliknya.

MS Comm Control 6.0 ini berfungsi untuk :

1. Mengadakan hubungan dengan *serial port PC*
2. Berhubungan dengan alat komunikasi lain (contoh : modem)
3. Melakukan pertukaran data
4. Memonitor dan merespon *event* dan *error* yang terjadi pada hubungan *serial*

Untuk melakukan sebuah sambungan komunikasi *serial* antara 2 peralatan (PC-IED atau PC-PC), harus dilakukan langkah – langkah sebagai berikut :

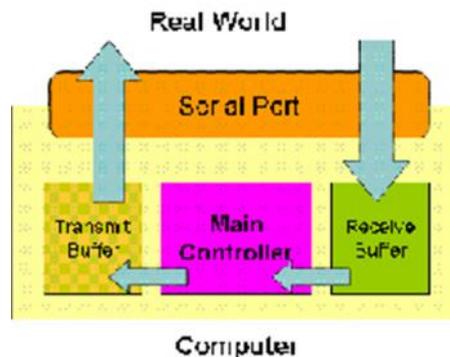
1. Membuka *serial port*
2. Mengatur *serial device*
3. Mengatur properti *Receive* dan *Transmit Buffer*
4. Mengatur manajemen *Receive* dan *Transmit Buffer*

Penjelasan masing – masing langkah akan diberikan di bawah.

1. Membuka *Serial Port*

Pada komunikasi serial, bit – bit data yang masuk dari dunia luar ke dalam komputer melalui *serial port* akan ditampung dulu di *receive buffer* sebelum akan dieksekusi oleh *main controller*. Demikian pula sebelum dikirim ke luar, data akan ditampung dulu di *transmit buffer*.

Skema lengkapnya dapat di lihat pada gambar 2.29.



Gambar 2.29. Skema Komunikasi MSComm

Sumber : <http://ari-sty-blog.blogspot.co.id/2009/11/antarmuka-ied-dengan-pc-melalui-mscomm.html>

Sebelum membuka *serial port*, dilakukan pengaturan protokol komunikasi *serial* dengan properti *MSComm* berikut :

- a) *CommPort* : menentukan nomor port komunikasi
- b) *Settings* : menentukan *baud rate*, *parity*, *data bits*, *stop bits* dalam *string*

Untuk membuka *serial port* cukup dengan properti :

PortOpen : membuka dan menutup *port*

Sehingga kode program akan tertulis sebagai berikut :

```
MSComm1.ComPort = 2
```

```
MSComm1.Settings = "9600,N,8,1"
```

```
MSComm1.PortOpen = True
```

2. Mengatur *Serial Device*

Pada tahap ini kita perlu memastikan bahwa pengaturan protokol komunikasi *serial* yang digunakan pada peralatan lain yang kita akses (misal : PLC, modem, mikrokontroler) sesuai dengan pengaturan pada komputer yang kita pakai.

3. Mengatur properti *Receive* dan *Transmit Buffer*

Ada beberapa properti dari *receive buffer* dan *transmit buffer* (properti dari *MS Comm*) yang perlu kita atur.

- a) *InBufferSize* : mengatur ukuran *receive buffer*
- b) *OutBufferSize* : mengatur ukuran *transmit buffer*
- c) *Rthreshold* : menentukan jumlah karakter yang diterima oleh *receive buffer* sebelum *OnComm event* dipicu
- d) *Sthreshold* : menentukan jumlah karakter yang diterima oleh *transmit buffer* sebelum *OnComm event* dipicu. Jika bernilai 0 berarti tidak pernah dipicu, sedangkan jika bernilai 1 berarti dipicu setiap satu karakter.
- e) *InputLen* : menentukan jumlah karakter yang dibaca CPU dari *receive buffer*. Jika bernilai " 0 ", maka seluruh isi *receive buffer* akan dibaca CPU.
- f) *InputMode* : menentukan tipe data *input* yang akan dibaca CPU. Jika *comInputModeText* yang dipilih maka *input mode* untuk data *string*/teks, sedangkan jika *comInputModeBinary* yang dipilih maka *input mode* untuk data *biner*.

4. Mengatur manajemen *Receive* dan *Transmit Buffer*

Untuk menampilkan data dari IED ke dalam aplikasi *visual basic*, digunakan properti *input*. Sehingga kode akan berbentuk :

```
TxtDisplay.Text = MSComm1.Input
```

Sedangkan untuk mengirim data dari aplikasi *visual basic* ke IED digunakan properti *output*. Sehingga kode akan berbentuk :

MSComm1.Output = "<nilai string>"

Untuk mengawasi jumlah bit yang ada di *transmit buffer* dan *receive buffer*, maka properti berikut dapat ditampilkan.

- a) *InBufferCount*
- b) *OutBufferCount*

MSComm hanya memiliki 1 *event*, yaitu *OnComm*. *Event* ini akan terjadi jika properti dari *CommEvent* berubah, yaitu saat terjadi :

- a) *Event* komunikasi
- b) *Error*

Contoh dari *event* komunikasi *OnComm* :

- a) *comEvSend* : mengirim sejumlah karakter ke *transmit buffer*
- b) *comEvReceive* : menerima sejumlah karakter di *receive buffer*

Sedang contoh dari error pada *OnComm* :

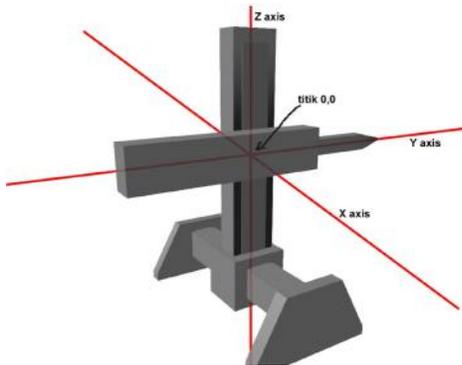
- a) *comEventBreak* : sinyal *break* diterima
- b) *comEventFrame* : ada kesalahan *framing*
- c) *comEventRxOver* : *receive buffer* mengalami *overflow*
- d) *comEventTXFull* : *transmit buffer* penuh¹⁵

¹⁵ Ari Sulistiono, Antarmuka IED dengan PC melalui MsComm VB6.0, <http://ari-sty-blog.blogspot.co.id/2009/11/antarmuka-ied-dengan-pc-melalui-mscomm.html>, 2009, diakses pada 4 Juli 2015.

2.3. Konfigurasi Manipulator Robot Kartesian

Lengan robot dapat melakukan berbagai macam gerakan untuk dapat menempatkan *end of effector* pada setiap titik di dalam ruang kerjanya. Kombinasi dari berbagai macam gerakan ini akan menentukan konfigurasi geometris lengan robot.

Pada robot manipulator konfigurasi kartesian, tiga buah *joint* pada lengan semuanya adalah *prismatic joint*. Gerak tiga derajat kebebasan pada lengan merupakan gerakan linier (translasi) yang saling tegak lurus dan membentuk daerah kerja berupa balok. Contohnya ditunjukkan oleh gambar 2.30.



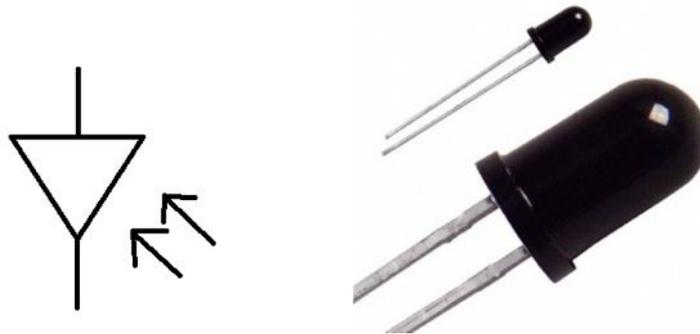
Gambar 2.30. Konfigurasi Kartesian

Sumber : <http://operator-it.blogspot.co.id/2013/11/mengenal-dasar-teknologi-robot.html>

2.4. Sensor Cahaya

Fotodioda adalah suatu dioda yang akan mengalami perubahan resistansi pada terminal anoda dan katoda apabila terkena cahaya. Sensor cahaya fotodioda merupakan sensor cahaya dengan tipe fotokonduktif, karena perubahan resistansi pada terminal *output*-nya sesuai dengan perubahan intensitas cahaya yang

diterimanya. Nilai resistansi anoda dan katoda pada fotodiode akan semakin rendah apabila intensitas cahaya yang diterima fotodiode semakin tinggi. Simbol dan bentuk fisik dari fotodiode dapat dilihat pada gambar 2.31.¹⁶



Gambar 2.31. Fotodiode

Sumber : <http://zoniaelektro.net/sensor-cahaya/>

Fotodiode memiliki hambatan sekitar 15 s/d 20 MOhm jika tidak terkena sinar infra merah, dan hambatannya akan berubah menjadi sekitar 80 s/d 300 KOhm jika terkena sinar infra merah tergantung dari besarnya intensitas yang mengenainya. Semakin besar intensitasnya, maka hambatannya semakin kecil.

Sistem sensor cahaya pada dasarnya menggunakan LED infra merah sebagai media untuk komunikasi data antara *receiver* dan *transmitter*. Sistem akan bekerja jika sinar infra merah yang dipancarkan terhalang oleh suatu benda yang mengakibatkan sinar tersebut tidak dapat terdeteksi oleh penerima. Keuntungan atau manfaat dari sistem ini dalam penerapannya antara lain sebagai pengendali jarak jauh, alarm keamanan, serta otomatisasi pada sistem. Pemancar pada sistem ini terdiri atas sebuah LED yang dilengkapi dengan rangkaian yang mampu membangkitkan data untuk dikirimkan melalui sinar tertentu, sedangkan pada

¹⁶ ZonaElektro, *Sensor Cahaya*, <http://zoniaelektro.net/sensor-cahaya/>, 2014, diakses pada 26 Mei 2015.

bagian penerima biasanya terdapat foto transistor, fotodiode, atau inframerah *module* yang berfungsi untuk menerima sinar tertentu yang dikirimkan oleh pemancar.

2.5. Radio Frequency Identification (RFID)

RFID merupakan sistem identifikasi dengan gelombang radio. Untuk penggunaannya minimal dibutuhkan dua buah perangkat, yaitu perangkat yang disebut RFID *tag* dan RFID *reader*. Saat pemindaian data, RFID *reader* membaca sinyal yang diberikan oleh RFID *tag*. RFID menawarkan keunggulan dibandingkan dengan sistem manual atau penggunaan *barcode*. *Tag* dapat dibaca jika lewat di dekat pembaca, bahkan jika itu ditutupi oleh objek atau tidak terlihat. *Tag* dapat dibaca dalam wadah, karton, kotak atau lainnya, dan tidak seperti *barcode*, RFID *tag* dapat sekaligus dibaca ratusan *id* pada suatu waktu menggunakan *reader* tertentu sedangkan *barcode* hanya dapat dibaca satu per satu menggunakan perangkat saat ini.

2.5.1. RFID tag

RFID *tag* adalah sebuah alat yang melekat pada obyek yang akan diidentifikasi oleh RFID *reader*. RFID *tag* dapat berupa perangkat pasif atau aktif. *Tag* pasif artinya tanpa baterai dan *tag* aktif artinya menggunakan baterai. RFID *tag* dapat berupa perangkat *read-only* yang berarti hanya dapat dibaca saja ataupun perangkat *read-write* yang berarti dapat dibaca dan ditulis ulang untuk keperluan pembaruan. RFID *tag* tidak berisi informasi pengguna

seperti nama, nomor rekening, NIK atau yang lain. RFID *tag* hanya berisi sebuah *tag* yang unik yang berbeda satu dengan yang lainnya. Jadi Informasi mengenai obyek yang terhubung ke tag ini hanya diperoleh pada sistem atau *database* yang terhubung pada RFID *reader*. Contoh bentuk fisik RFID *tag* ditunjukkan pada gambar 2.32.



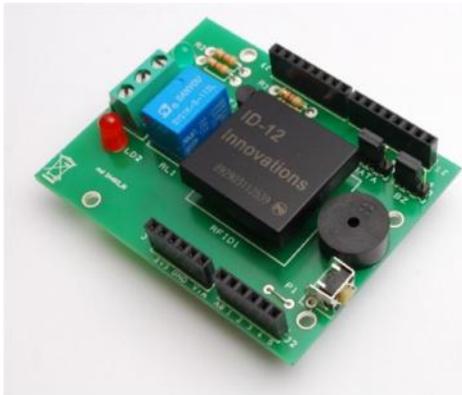
Gambar 2.32. Contoh RFID tag

Sumber: <http://green-elektronik.blogspot.co.id/2010/07/RFID.html>

2.5.2. RFID reader

RFID *reader* merupakan alat pembaca RFID *tag*. Ada dua macam RFID *reader* yaitu *reader* pasif (PRAT) dan *reader* aktif (ARPT). *reader* pasif memiliki sistem pembaca pasif yang hanya menerima sinyal radio dari RFID *tag* aktif (yang dioperasikan dengan baterai/sumber daya). Jangkauan penerima RFID pasif bisa mencapai 600 meter. Hal ini memungkinkan aplikasi RFID untuk sistem perlindungan dan pengawasan aset. *Reader* aktif memiliki sistem pembaca aktif yang memancarkan sinyal interogator ke *tag* dan menerima balasan autentikasi dari *tag*. Sinyal interogator ini juga menginduksi *tag* dan akhirnya menjadi sinyal DC yang menjadi sumber daya

tag pasif. Contoh bentuk fisik dari RFID *reader* dapat dilihat pada gambar 2.33.¹⁷



Gambar 2.33. RFID Reader ID-12

Sumber : <http://www.rlocman.ru/i/Image/2012/05/31/3.jpg>

2.6. Motor DC Magnet Permanen

Motor DC (*direct curent*) adalah peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik yang disain awalnya diperkenalkan oleh Michael Faraday lebih dari seabad yang lalu.¹⁸ Tenaga atau energi mekanik di sini umumnya gerak rotasi dengan memanfaatkan induksi magnetik yang dihasilkan oleh energi listrik. Hal tersebut bersesuaian dengan gaya Lorentz dan kaidah tangan kiri Flemming seperti pada gambar 2.34. Ketika sebuah konduktor beraliran arus diletakkan dalam medan magnet, maka sebuah gaya (yang dikenal dengan gaya Lorentz) akan tercipta secara ortogonal diantara arah medan magnet dan arah aliran arus. Atau dapat dianalogikan telapak tangan

¹⁷ Eko Purnomo, *Prinsip Kerja RFID*, <https://abisabrina.wordpress.com/2014/01/18/prinsip-kerja-rfid/#more-2045>, diakses pada 5 Juli 2015

¹⁸ Endra Pitowarno. *Robotika: Desain, Kontrol, dan Kecerdasan Buatan*. (Yogyakarta: Andi. 2006) hal.76

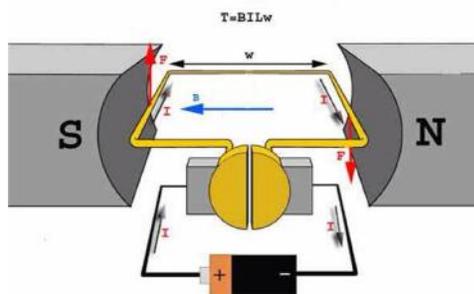
menghadap kutub utara, di mana arah medan magnet menuju ke telapak tangan. Arah gerak kawat ditunjukkan oleh ibu jari, dan arah 4 (empat) jari yang lain menunjukkan arah arus yang mengalir.



Gambar 2.34. Kaidah Tangan Kiri Flemming

Sumber : Dokumen pribadi

Pada motor DC terdapat jangkar dengan satu atau lebih kumparan terpisah. Tiap kumparan berujung pada cincin belah (komutator). Dengan adanya insulator antara komutator, cincin belah dapat berperan sebagai saklar kutub ganda (*double pole, double throw switch*). Seperti ditunjukkan pada gambar 2.35.



Gambar 2.35. Mekanisme Kerja Motor DC Magnet Permanen dengan Cincin Belah

Sumber : <http://artikel-teknologi.com/prinsip-kerja-motor-listrik/>

Motor DC yang digunakan pada robot beroda umumnya adalah motor DC dengan magnet permanen. Motor DC jenis ini memiliki dua buah magnet

permanen sehingga timbul medan magnet di antara kedua magnet tersebut. Di dalam medan magnet inilah jangkar/rotor berputar. Jangkar yang terletak di tengah motor memiliki jumlah kutub yang ganjil dan pada setiap kutubnya terdapat lilitan. Lilitan ini terhubung ke area kontak yang disebut komutator. Sikat (*brushes*) yang terhubung ke kutub positif dan negatif motor memberikan daya ke lilitan sedemikian rupa sehingga kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya, sedangkan lilitan lain akan ditarik ke magnet permanen yang lain sehingga menyebabkan jangkar berputar. Ketika jangkar berputar, komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga jangkar akan terus berputar selama kutub positif dan negatif motor diberi daya.

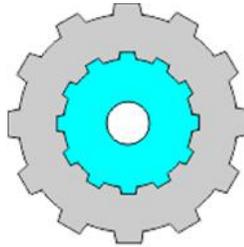
2.7. Gear Box (Hubungan Roda-roda)

Pada dalam *gear box* ada beberapa kemungkinan hubungan gir, kemungkinan-kemungkinan tersebut yaitu dua roda pada poros yang sama, dua roda dihubungkan dengan satu rantai, dan roda saling bersinggungan.

2.7.1. Dua Roda dengan Poros yang Sama

Pada kondisi ini dua roda berbeda ukuran berada pada satu poros yang sama seperti yang ditunjukkan pada gambar 2.36. Akibatnya kedua roda mempunyai kecepatan sudut yang sama dengan arah yang sama. Karena panjang jari-jari roda berbeda, ada yang besar ada yang kecil maka kecepatan liniernya berbeda. Semakin besar ukuran (jari-jari) roda maka akan semakin

besar kecepatan liniernya. Rumus persamaannya ditunjukkan oleh rumus 2.3, 2.4, dan 2.5.



Gambar 2.36. Dua Roda pada Poros yang Sama

Sumber : <http://rumushitung.com/2013/07/12/hubungan-roda-roda/>

$$\omega_a = \omega_b \dots\dots\dots (2.3)$$

$$v = \omega \cdot r \dots\dots\dots (2.4)$$

Maka :

$$\frac{v_a}{r_a} = \frac{v_b}{r_b} \dots\dots\dots (2.5)$$

Keterangan :

v_a = kecepatan linier roda a (m/s)

v_b = kecepatan linier roda b (m/s)

ω_a = kecepatan sudut roda a (rad/s)

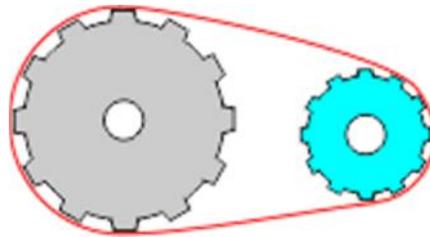
ω_b = kecepatan sudut roda b (rad/s)

r_a = jari-jari roda a (m)

r_b = jari-jari roda b (m)

2.7.2. Dua Roda Dihubungkan Satu Rantai

Ketika dua roda kita hubungkan dengan sebuah rantai seperti ditunjukkan pada gambar 2.37, maka kedua roda tersebut akan memiliki kecepatan linier yang sama (sama dengan kecepatan gerak rantai). Tidak hanya besar kecepatan liniernya yang sama tapi juga arah dari gerakan roda. Dalam hubungan roda ini yang berbeda adalah kecepatan sudutnya.



Gambar 2.37. Dua Roda Dihubungkan Satu Rantai

Sumber : <http://rumushitung.com/2013/07/12/hubungan-roda-roda/>

$$v_a = v_b \dots\dots\dots (2.5)$$

$$\omega_a \cdot r_a = \omega_b \cdot r_b \dots\dots\dots (2.6)$$

$$\frac{\omega_a}{r_b} = \frac{\omega_b}{r_a} \dots\dots\dots (2.7)$$

Atau bisa juga ditulis :

$$\frac{n_a}{d_b} = \frac{n_b}{d_a} \dots\dots\dots (2.8)$$

Di mana :

n_a = kecepatan putaran roda a (rpm)

n_b = kecepatan putaran roda b (rpm)

d_a = diameter roda a (m)

d_b = diameter roda b (m)

Jika kita menginginkan kecepatan putaran roda kedua tiga kali lebih cepat dari kecepatan putaran roda pertama, maka dapat disiasati dengan mengecilkan ukuran diameter roda kedua tiga kali lebih kecil ukurannya dari diameter roda pertama. Misal kita menginginkan kecepatan putaran roda b sebesar 1500 rpm. Sedangkan roda a dengan diameter roda 30 cm memiliki kecepatan putaran sebesar 500 rpm. Maka diameter roda b dapat ditentukan dengan rumus menggunakan rumus 2.8 dengan perhitungan berikut.

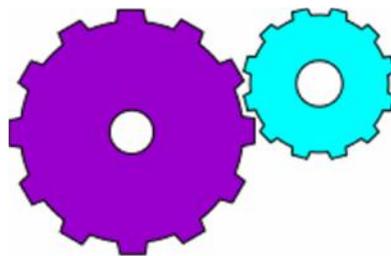
$$\frac{5}{d_b} = \frac{1}{0,3}$$

$$d_b = 0,1 \text{ m} = 10 \text{ cm}$$

Jadi untuk mendapatkan putaran 1500 rpm maka dibutuhkan diameter roda sebesar 10cm.

2.7.3. Roda Saling Bersinggungan

Pada kondisi ini roda saling bersinggungan satu sama lain. Seperti pada gambar 2.38. Sistem hubungan roda ini mirip dengan poin 2.8.2 tetapi yang membedakannya adalah arah kecepatan liniernya yang berbeda. Jadi kecepatan linier sama tapi berbeda arah dan kecepatan sudutnya berbeda.



Gambar 2.38. Dua Roda Saling Bersinggungan

Sumber : <http://rumushitung.com/2013/07/12/hubungan-roda-roda/>

$$v_a = -v_b \dots\dots\dots (2.9)$$

Tanda (-) menunjukkan bahwa roda berputar berlawanan arah.

Penjabaran tersebut merupakan hubungan antara kecepatan sudut, kecepatan linear dengan hubung roda. Sedangkan untuk hubungannya dengan torsi dan diameter roda dapat dijabarkan melalui rumus :

$$\tau = F \cdot l \dots\dots\dots (2.7)$$

Di mana :

τ = torsi atau momen gaya (Nm)

F = gaya gerak (N)

l = lengan gaya atau bisa dikatakan jari-jari roda (m)

Pada roda yang berhubungan tersebut bisa dikatakan bahwa nilai F pada roda a sama dengan nilai F pada roda b. Sehingga :

$$\frac{\tau_1}{l_1} = \frac{\tau_2}{l_2} \dots\dots\dots (2.8)$$

Di mana :

τ_1 = torsi pada roda 1 (Nm)

τ_2 = torsi pada roda 2 (Nm)

l_1 = lengan gaya roda 1 (m)

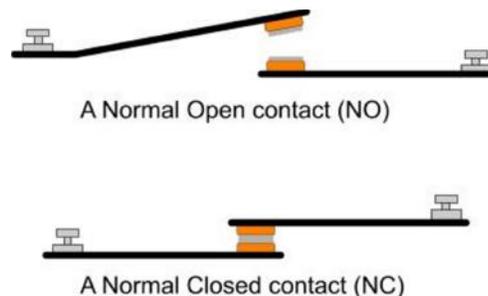
l_2 = lengan gaya roda 2 (m)

Jadi perbandingan jari-jari pada roda dapat menentukan besar torsi dan kecepatan berdasarkan rumus-rumus di atas.

2.8. Saklar Pembatas (*Limit Switch*)

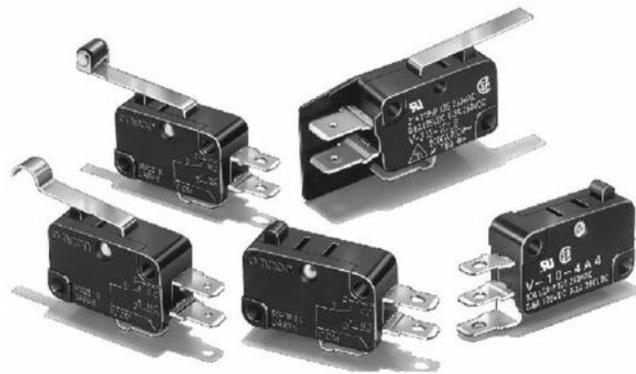
Saklar pembatas umumnya digunakan di dunia industri, saklar pembatas dirancang untuk beroperasi secara otomatis apabila batas yang diinginkan sudah dicapai dan saklar diaktifkan secara mekanik. Jadi saklar pembatas secara tidak langsung berfungsi sebagai pengganti operator.

Saklar pembatas dioperasikan secara mekanik, artinya saklar pembatas dikontrol oleh faktor-faktor lain secara otomatis, misal : tekanan dan posisi. Gambar 2.33 bagian atas menunjukkan kontak *normally open* (NO), rangkaian akan terbuka (tidak terhubung) ketika tuas kontak tidak ditekan dan akan tertutup (terhubung) ketika tuas kontak ditekan. Gambar 2.39 bagian bawah menunjukkan kontak *normally close* (NC), pada komponen ini rangkaian akan tertutup (terhubung) ketika tuas kontak tidak ditekan dan akan terbuka (tidak terhubung) ketika tuas kontak ditekan. Sedangkan untuk contoh fisik dari *limit switch* ditunjukkan oleh gambar 2.40.



Gambar 2.39. Kontak-kontak *Limit Switch*

Sumber : http://pcbheaven.com/wikipages/How_Relais_Work/

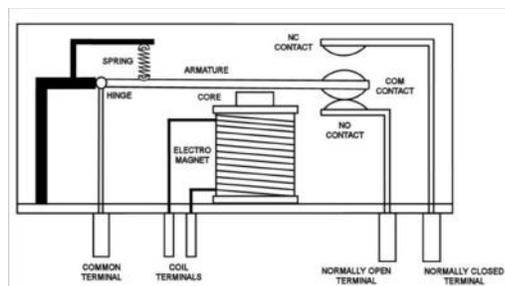


Gambar 2.40. Bentuk Fisik Macam-macam *Limit Switch*

Sumber : <http://abccoolimages.com/miniatuure+limit+switch>

2.9. Relai

Relai pengendali elektromekanis (*electromechanical relai* = EMR) adalah saklar magnetis¹⁹. Dengan kata lain menggunakan elektromagnet untuk mengoperasikan seperangkat kontak sakelar. Susunan paling sederhana terdiri dari kumparan kawat penghantar yang dililit pada inti besi. Bila kumparan ini dienergikan, medan magnet yang terbentuk menarik armatur berporos yang digunakan sebagai pengungkit mekanisme sakelar. Sebagai contoh dapat dilihat pada gambar 2.41.

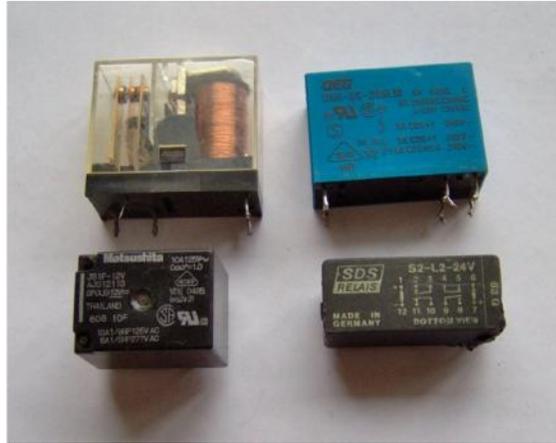


Gambar 2.41. Rangkaian Internal Relai

Sumber : <http://elektronika-dasar.web.id/teori-elektronika/teori-relai-elektro-mekanik/>

¹⁹ Frank D. Petruzella, *Elektronika Industri. Terj*, Andi, Yogyakarta, 2001, hlm. 371

Relai merupakan bagian penting dari banyak tipe sistem kendali, bermanfaat sebagai pengendali jarak jauh, pengendali alat dengan tegangan dan arus tinggi dengan sinyal kendali tegangan dan kendali arus yang rendah. Bentuk fisik relai dapat dilihat pada gambar 2.42.



Gambar 2.42. Macam-macam relai

Sumber: https://upload.wikimedia.org/wikipedia/commons/8/8a/Electronic_component_relais.jpg

2.10. Solenoid Doorlock

Solenoid door lock adalah pengunci pintu otomatis yang dapat dikontor oleh tegangan listrik. Tegangan *input* untuk *doorlock* biasanya berkisar antara 12-24V DC.²⁰ *Solenoid* merupakan kumparan terbuat dari konduktor yang dililitkan secara rapat. Dimana medan magnet yang dihasilkannya seragam dan paralel terhadap sumbu solenoid. Prinsip kerja *solenoid doorlock* adalah ketika besi pengunci yang ditempatkan sebagian panjangnya di dalam solenoid, lalu solenoid dialiri listrik

²⁰ Anonim, *Solenoid Doorlock*, <http://www.geraicerdas.com/motor/solenoid-door-lock-detail>, diakses pada tanggal 3 Oktober 2015

maka batang tersebut akan bergerak masuk ke dalam solenoid sehingga kunci terbuka.²¹ Bentuk fisik dari solenoid ditunjukkan oleh gambar 2.43.



Gambar 2.43. Solenoid Doorlock

Sumber : <http://www.geraicerdas.com/motor/solenoid-door-lock-detail>

2.11. Kerangka Berfikir

Pada *Prototype* penyimpanan benda bertingkat otomatis ini menggunakan PLC dan aplikasi *windows* yang dibuat menggunakan *visual basic 6.0* sebagai pusat kerja penyimpanannya. Selain menyimpan secara otomatis *prototype* ini juga disertai dengan pengaman RFID dan *password* untuk penyimpanan dan pengambilan benda, serta akses khusus administrator ketika ada hal-hal yang tidak diinginkan seperti lupa *password* terjadi. Hal tersebut agar penyimpanan menjadi lebih aman. Secara garis besar terdapat 3 bagian utama penyusun *prototype* ini yaitu *input*, proses, dan *output*.

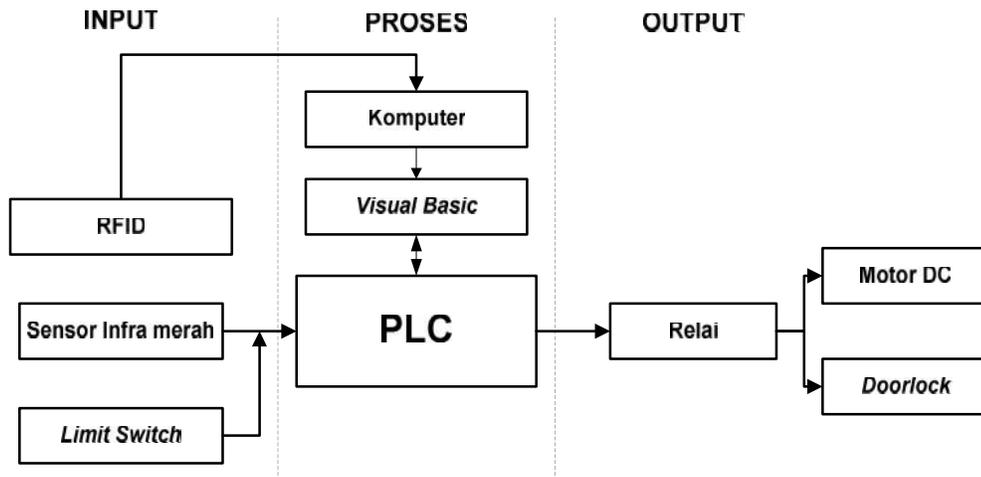
Sinyal *input* yang digunakan oleh PLC ada 12 buah *limit switch*, 1 buah sensor infra merah, RFID yang terhubung melalui aplikasi *visual basic* serta aplikasi itu sendiri. *Input limit switch* digunakan untuk mendeteksi pergerakan pemindah benda. Apabila sudah mencapai batas tertentu *limit switch* akan tertekan dan

²¹ Anonim, *Solenoid*, <https://id.wikipedia.org/wiki/Solenoid>, diakses pada tanggal 3 Oktober 2015

memberikan sinyal ke PLC. Sehingga pemindah benda tidak melebihi *degree of movement* yang sudah ditentukan. Sedangkan *input* sensor infra merah digunakan untuk mendeteksi benda yang akan disimpan atau diambil berada pada pemindah benda.

Pengolah proses sistem penggerak pada alat ini secara keseluruhan adalah PLC Omron CJ1M. Di mana PLC memproses setiap sinyal secara otomatis berdasarkan diagram *ladder* yang sudah dibuat. sinyal *input* yang masuk kemudian diproses oleh PLC dan dilanjutkan dengan memberi sinyal *output* yang dikehendaki. Untuk pengolah proses sistem keamanan dan *interface* pada alat ini diproses sepenuhnya oleh aplikasi *visual basic*. Sehingga antara PLC CJ1M dan aplikasi *visual basic* saling terintegrasi satu sama lain.

Untuk sinyal *output* pada *prototype* ini sebagian besar terhubung kepada relai, yang pada akhirnya relai ini berfungsi untuk mengontrol daya dan arah putaran dari motor DC penggerak *axis x*, *axis y*, *axis z*, dan pintu depan, serta daya pada *doorlock*. Sebagian *output* terkoneksi pada aplikasi *visual basic*. Dimana *output* tersebut berfungsi sebagai indikator apakah *doorlock* dalam keadaan aktif atau tidak. Untuk lebih jelasnya hubungan antara komponen *input*, proses dan *output* dari alat ini dapat dilihat pada gambar 2.44.



Gambar 2.44. Diagram Blok Alat

Sumber : Dokumen Pribadi

BAB III

METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penelitian dilakukan di bengkel kerja di luar kampus Universitas Negeri Jakarta, serta Laboratorium PLC, dan Bengkel Mekanik Program Studi Teknik Elektro Fakultas Teknik Universitas Negeri Jakarta yang dilaksanakan pada bulan Juli tahun 2015 sampai dengan bulan Desember tahun 2015.

3.2. Metode Penelitian

Metode yang digunakan pada penelitian pembuatan *prototype* penyimpanan benda bertingkat otomatis dengan pengaman *password* dan RFID berbasis PLC Menggunakan *Human Machine Interface* adalah metode eksperimen laboratorium. Langkah awal dalam penelitian ini dimulai dengan membuat rancangan alat dan rancangan *software* terlebih dahulu. Kemudian dilanjutkan dengan pembuatan alat yang dilanjutkan dengan perbaikan *software* agar ada penyesuaian dengan alat yang sudah terbentuk. Dan pada tahap akhir dilanjutkan dengan pengujian alat.

3.3. Rancangan Penelitian

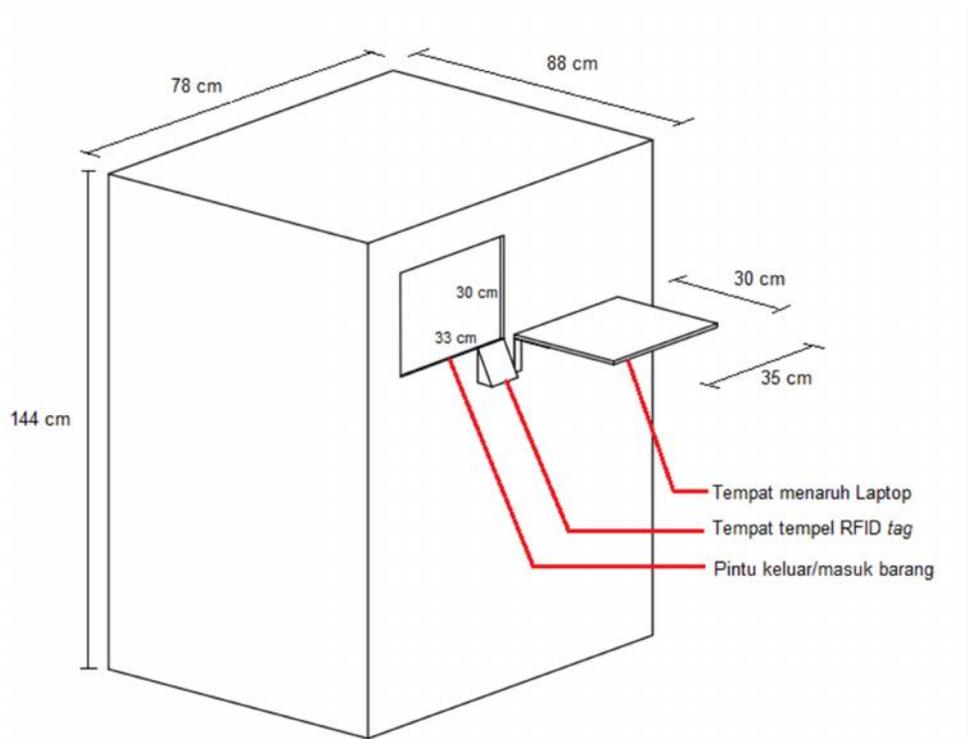
3.3.1. Perancangan Mekanik Penyimpanan Benda Bertingkat Otomatis

Prototype alat ini berukuran cukup besar, yaitu memiliki tinggi 144cm, panjang 88cm dan lebar 78cm. Di bagian depan *prototype* terdapat pintu

keluar masuk barang, tempat menempel RFID dan tatakan untuk laptop. Ukuran pintu keluar masuk barang berukuran 33cm x 30cm. Ukuran tatakan laptop berukuran panjang 30cm dan lebar 35cm. Penyimpanan pada alat ini terdapat 3 tingkat. Dimana tiap 2 slot terdapat pada 1 tingkat. Sehingga total slot berjumlah 6 slot.

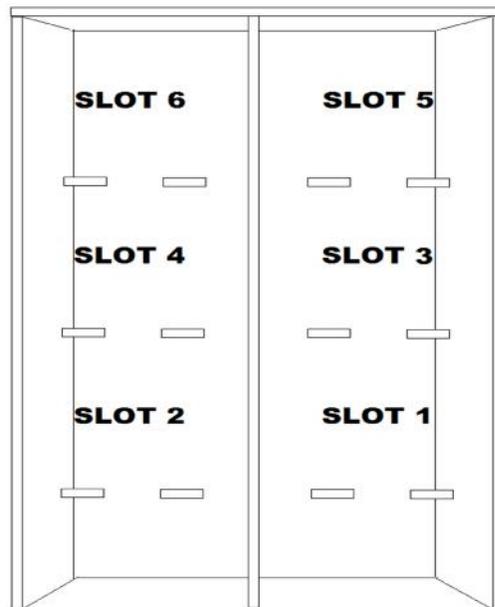
Alat ini menggunakan 4 buah motor 24 VDC dengan *gearbox* sebagai penggerak utamanya. 1 buah motor sebagai penggerak *axis X*, 1 buah motor sebagai penggerak *axis Y*, 1 buah motor sebagai penggerak *axis Z*, dan 1 buah motor sebagai penggerak pintu depan. Sistem pergerakannya dengan konfigurasi robot kartesian sehingga ruang kerja yang terbentuk berbentuk kubus, selain itu juga dengan konfigurasi kartesian memiliki ketelitian yang cukup tinggi.

Walaupun menerapkan sistem otomatis dalam menyimpan dan mengambil benda, alat ini juga masih memiliki pintu darurat pada bagian samping yang bisa dibuka dalam keadaan darurat oleh administrator. Perencanaan mekanik dari *prototype* penyimpanan benda bertingkat otomatis ditunjukkan oleh gambar 3.1, 3.2 dan 3.3. Gambar 3.1 menunjukkan tampilan luar penyimpanan benda, gambar 3.2 menunjukkan pemetaan slot penyimpanan benda, dan gambar 3.3 menunjukkan tampilan dalam penyimpanan benda.



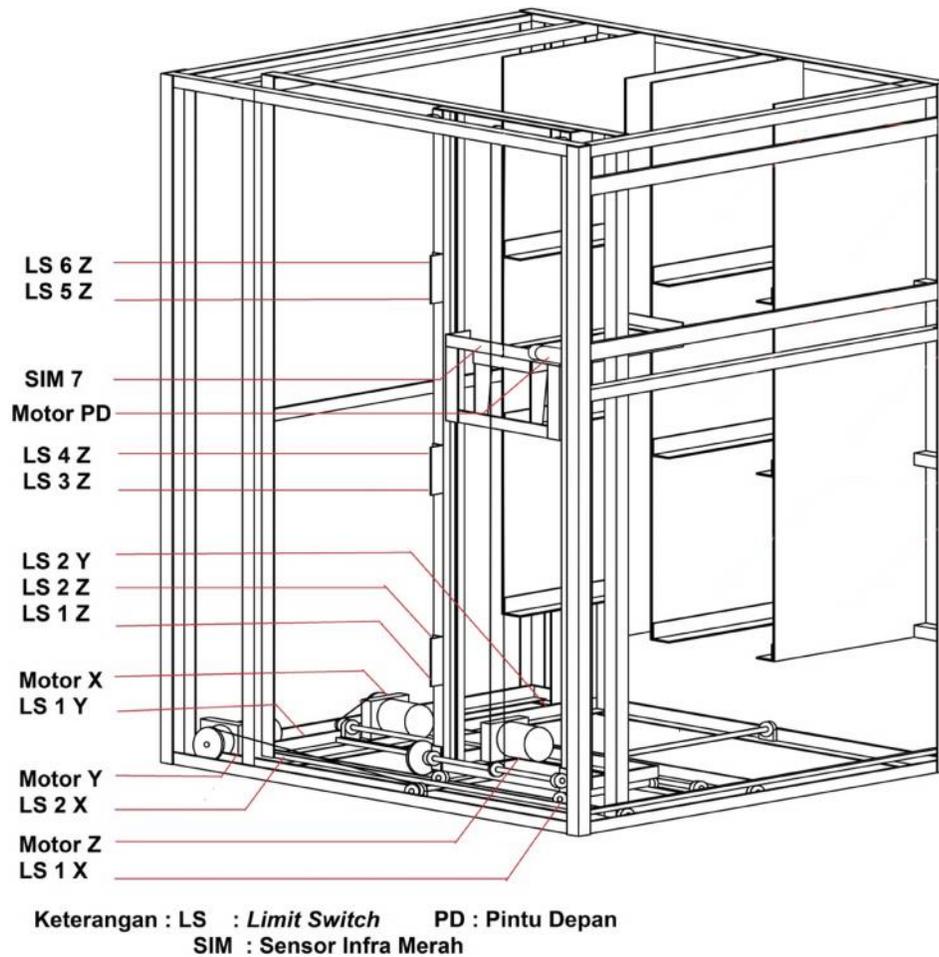
Gambar 3.1. Penyimpanan Tampak Luar

Sumber : Dokumen pribadi



Gambar 3.2. Pemetaan Slot

Sumber : Dokumen Pribadi

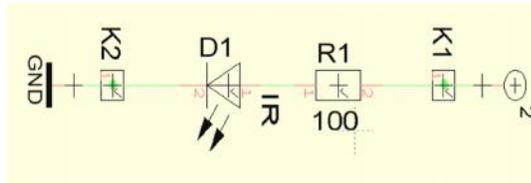


Gambar 3.3. Penyimpanan Tampak Dalam

Sumber : Dokumen pribadi

3.3.2. Perancangan sensor infra merah

Rangkaian sensor ini digunakan untuk memberi sinyal ada tidaknya benda pada tiap-tiap slot. Komponen rangkaian *transmitter* sensor hanya terdiri atas resistor, dan LED infra merah. Seperti yang ditunjukkan oleh gambar 3.4.



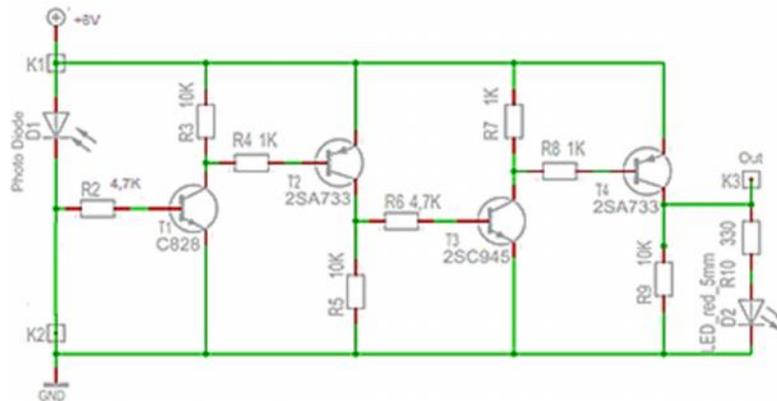
Gambar 3.4. Rangkaian Transmitter

Sumber : Dokumen pribadi

Pada rangkaian di atas digunakan sebuah LED infra merah yang diserikan dengan sebuah resistor 100 Ohm. Resistor ini berfungsi untuk membatasi arus yang masuk ke LED infra merah agar LED infra merah tidak rusak. Sehingga arus yang mengalir pada LED infra merah adalah sebesar:

$$i = \frac{V}{R} = \frac{6}{1} = 0,06 \text{ a} \quad 60m \dots\dots\dots(2.1)$$

Sedangkan untuk *receiver* sensor ini terdiri atas resistor, transistor, fotodiode, dan relai. Untuk lebih jelasnya dapat dilihat pada gambar 3.5.



Gambar 3.5. Rangkaian Receiver

Sumber : Dokumen pribadi

Pancaran dari sinar infra merah akan diterima oleh fotodiode, kemudian akan diolah oleh rangkaian penerima agar menghasilkan data *biner*, di mana jika fotodiode menerima pancaran sinar infra merah maka *output* dari rangkaian

penerima ini akan mengeluarkan logika *low* (0), namun jika photodiode tidak menerima pantulan sinar infra merah, maka *output* dari rangkaian penerima akan mengeluarkan logika *high* (1).

Pada rangkaian di atas, *output* dari fotodiode diumpankan ke basis dari transistor tipe NPN C828, ini berarti untuk membuat transistor tersebut aktif maka tegangan yang keluar dari fotodiode harus lebih besar dari 0,7 volt, syarat ini akan terpenuhi jika mendapat sinar infra merah. Analisisnya sebagai berikut:

Jika ada sinar infra merah yang mengenai fotodiode, maka hambatan pada fotodiode 300Kohm, sehingga:

$$V_b = \frac{R_2}{R_1+R_2} \times V_c \dots\dots\dots (2.2)$$

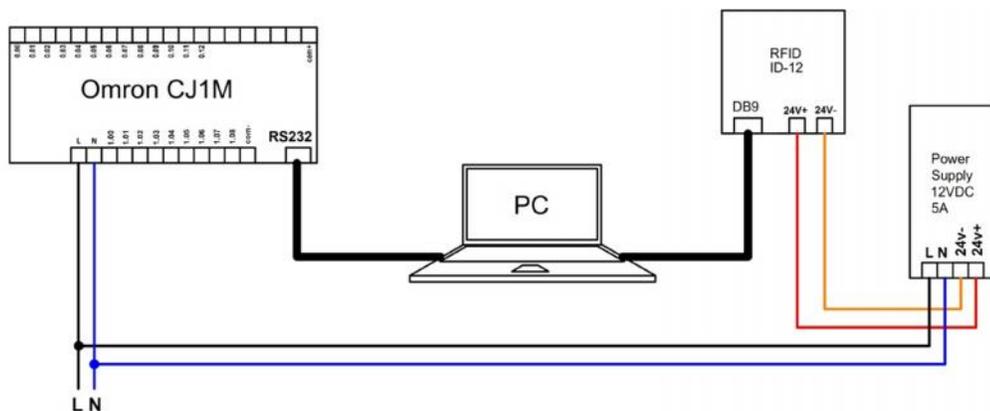
V_{out} akan diumpankan ke basis dari transistor C828, jika tegangannya lebih besar dari 0,7 volt maka transistor akan aktif. Aktifnya transistor C828 akan menyebabkan kolektornya terhubung ke emitor. Sehingga kolektor mendapat tegangan 0 volt, tegangan ini diumpankan ke basis dari transistor ke-2 tipe PNP A733, sehingga transistor ini juga aktif. Seterusnya aktifnya transistor A733 akan menyebabkan kolektor terhubung ke emitor, sehingga kolektor mendapat tegangan 6 volt dari V_{cc} . Tegangan ini juga diumpankan ke basis dari transistor ke 3 tipe NPN C945, sehingga transistor ini juga aktif.

Transistor ke 4 tipe PNP A733 berfungsi untuk mengaktifkan relai di mana kaki NC relai digunakan sebagai *input* PLC. *Input* ini akan aktif jika photodiode

tidak menerima sinar infra merah, dan akan mati jika photo dioda menerima sinar infra merah.

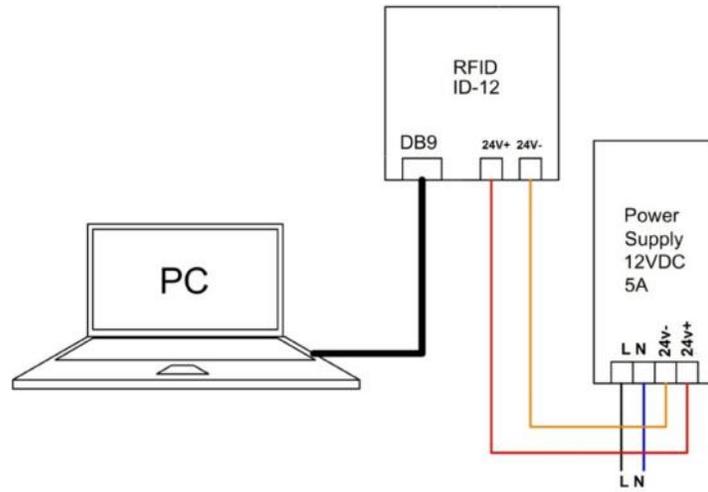
3.3.3. Perancangan *wiring*

Sedangkan untuk wiring diagram dari *prototype* penyimpanan benda bertingkat otomatis di bagi menjadi beberapa bagian, untuk komunikasi antara RFID, PC dan PLC ditunjukkan oleh gambar 3.6, untuk instalasi RFID ditunjukkan oleh gambar 3.7, untuk instalasi instalasi motor listrik ditunjukkan oleh gambar 3.8 sedangkan untuk instalasi pada panel alat ditunjukkan oleh gambar 3.9.



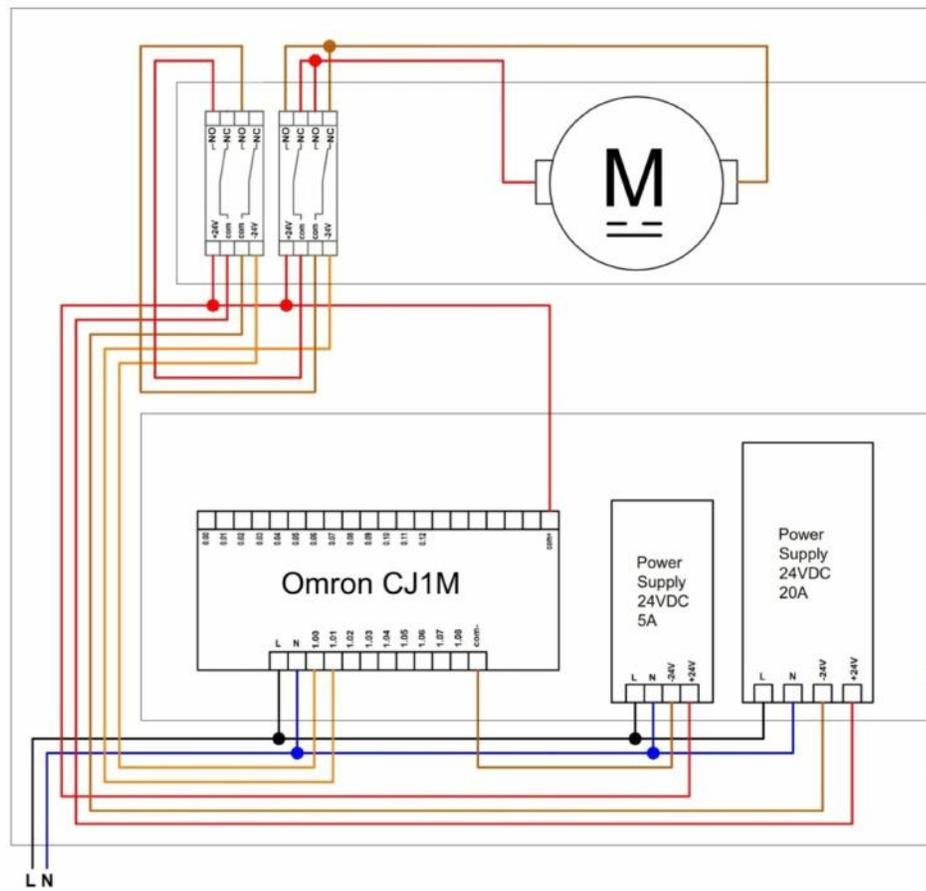
Gambar 3.6. Komunikasi PC, PLC, dan RFID

Sumber : Dokumen Pribadi



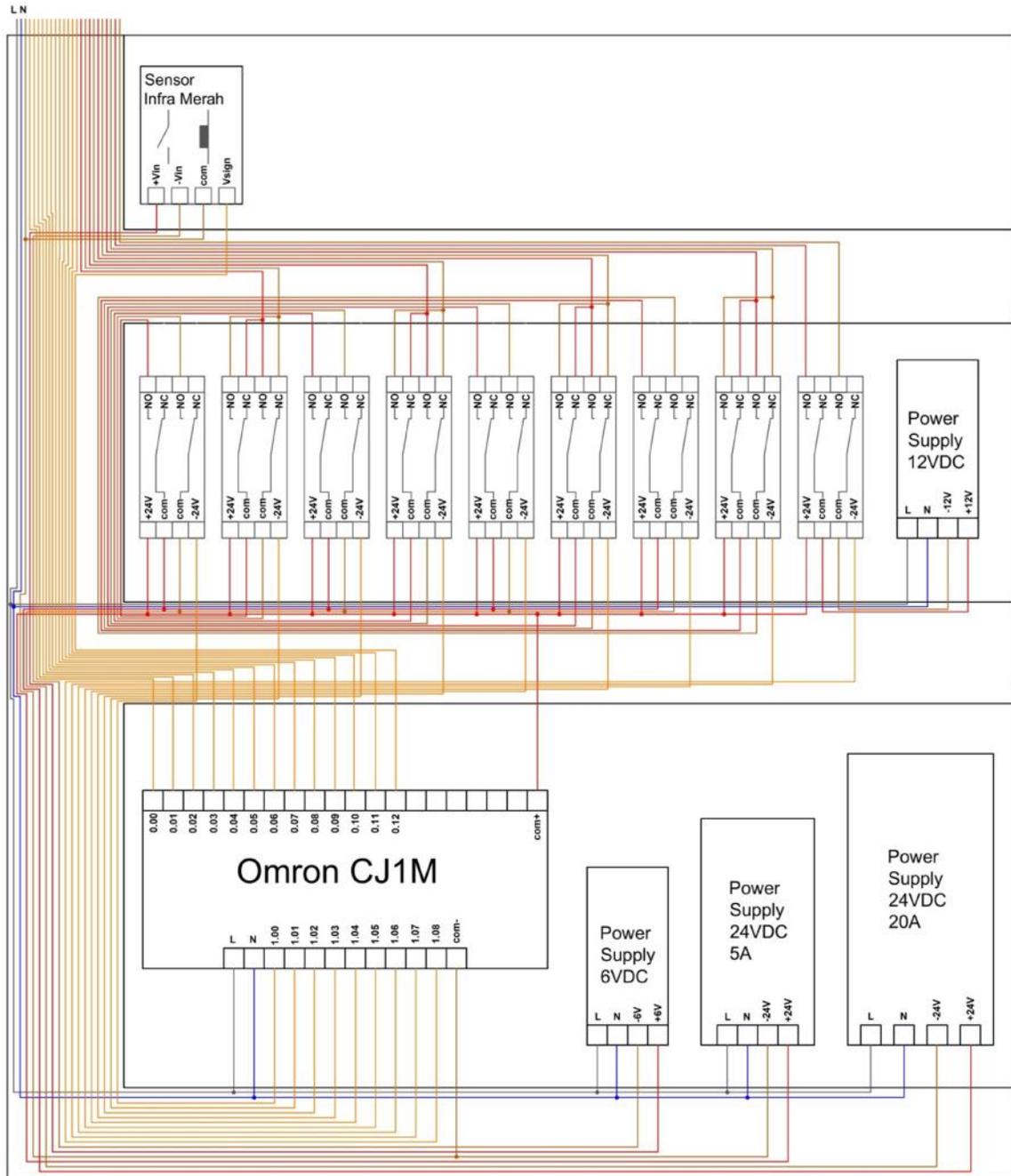
Gambar 3.7. Instalasi RFID

Sumber : Dokumen Pribadi



Gambar 3.8. Instalasi Motor Listrik

Sumber : Dokumen Pribadi



Gambar 3. 9. Wiring Diagram Prototype Panel Penyimpanan Benda Otomatis

Sumber : Dokumen Pribadi

3.3.4. Perancangan Program

Dalam merancang suatu program untuk membuat penyimpanan benda bertingkat otomatis ada beberapa tahap yang harus dilakukan, mulai dari pembuatan desain sistem secara kasar, deskripsi kerja sistem, pembuatan diagram *ladder* atau program PLC, pembuatan *software* menggunakan *visual basic*, hingga pengaturan komunikasi antara RFID, PC dan PLC.

3.3.4.1. Deskripsi Kerja Sistem

Untuk deskripsi sistem kerja alat dijelaskan sebagai berikut: Pada saat awal *software* yang dibuat melalui *visual basic* akan menampilkan tulisan “Selamat datang di peyimpanaan benda otomatis, silahkan tempelkan kartu Anda” di tampilan *form* menu pembuka. Untuk proses sistem pengaman, penyimpanan dan pengambilan benda dijelaskan pada poin berikut.

1) Sistem Pengaman

- a. Pada saat awal *software* akan membaca apakah terdapat input dari RFID. Jika tidak ada input tampilan *form* menu pembuka akan ditampilkan terus.
- b. Jika RFID *tag* pengguna terbaca oleh RFID *reader* maka *form* menu pembuka akan bereaksi yaitu Jika RFID *tag* 1 / RFID tag 2 / RFID tag 3 ditempelkan dan pengguna belum menentukan *password* maka *form* pengisian *password* baru akan muncul.

- c. Sedangkan jika RFID tag 1 / RFID tag 2 / RFID tag 3 ditempelkan dan pengguna sudah menentukan *password* maka *form* pengisian klarifikasi *password* akan muncul.
- d. Pada saat pengisian *password* baru *software* akan memverifikasi apakah *password* tersebut ketika dimasukan tidak mengalami salah ketik dengan cara meminta kembali pengguna untuk memasukan *password* yang sebelumnya dimasukan. Jika *password* pertama dan kedua sama maka *form* menu pengguna akan ditampilkan oleh *software*. Dimana pada menu tersebut terdapat pilihan untuk menyimpan benda atau mengambil benda serta pilihan untuk mengganti *password*.
- e. Pada saat menu klarifikasi *password* dimana pengguna sudah mengatur *password*-nya jika terjadi kesalahan input sebanyak tiga kali maka hak penggunaan alat oleh *software* akan disegel sementara. Dan pengguna harus menghubungi administrator untuk melakukan penggantian *password* dan pengembalian hak penggunaanya kembali. Hal ini berguna untuk mencegah ketika RFID tag hilang atau bukan berada di pengguna yang seharusnya, dan benda yang disimpan di penyimpanan otomatis ini tetap terjamin aman.
- f. Data pada saat pengguna menyimpan benda akan ditampilkan pada menu pengambilan benda di menu pengguna tersebut. Guna menginformasikan kapan pengguna menyimpan benda tersebut.

Data tersebut akan kembali dikosongkan setelah benda tersebut kembali diambil oleh pengguna.

- g. Data setiap *slot* mengenai kapan *benda* disimpan di *slot* tersebut serta siapa yang menyimpannya dapat dipantau pada menu administrator. Selain itu juga pada menu administrator terdapat pilihan untuk me-*reset* kembali *password* pengguna, pilihan untuk membuka pintu darurat, serta peringatan bahwa terjadi percobaan gagal *password* sebanyak tiga kali.

2) Penyimpanan Benda

- a. Pada *form* menu pengguna tekan tombol simpan untuk menyimpan benda.
- b. Menu *form* menyimpan benda muncul pada layar, pintu penyimpanan terbuka.
- c. Benda ditaruh di posisi awal, yang merupakan tempat sirkulasi benda masuk dan keluar.
- d. Pengguna memilih *slot* mana yang akan digunakan untuk menyimpan benda tersebut.
- e. Setelah memilih maka pintu penyimpanan akan kembali menutup dan benda akan disimpan di *slot* yang pengguna pilih sebelumnya.
- f. Ketika benda disimpan pada *slot* tertentu, misalnya *slot* 1, maka pilihan *slot* 1 pada pengguna lain akan menghilang sehingga yang

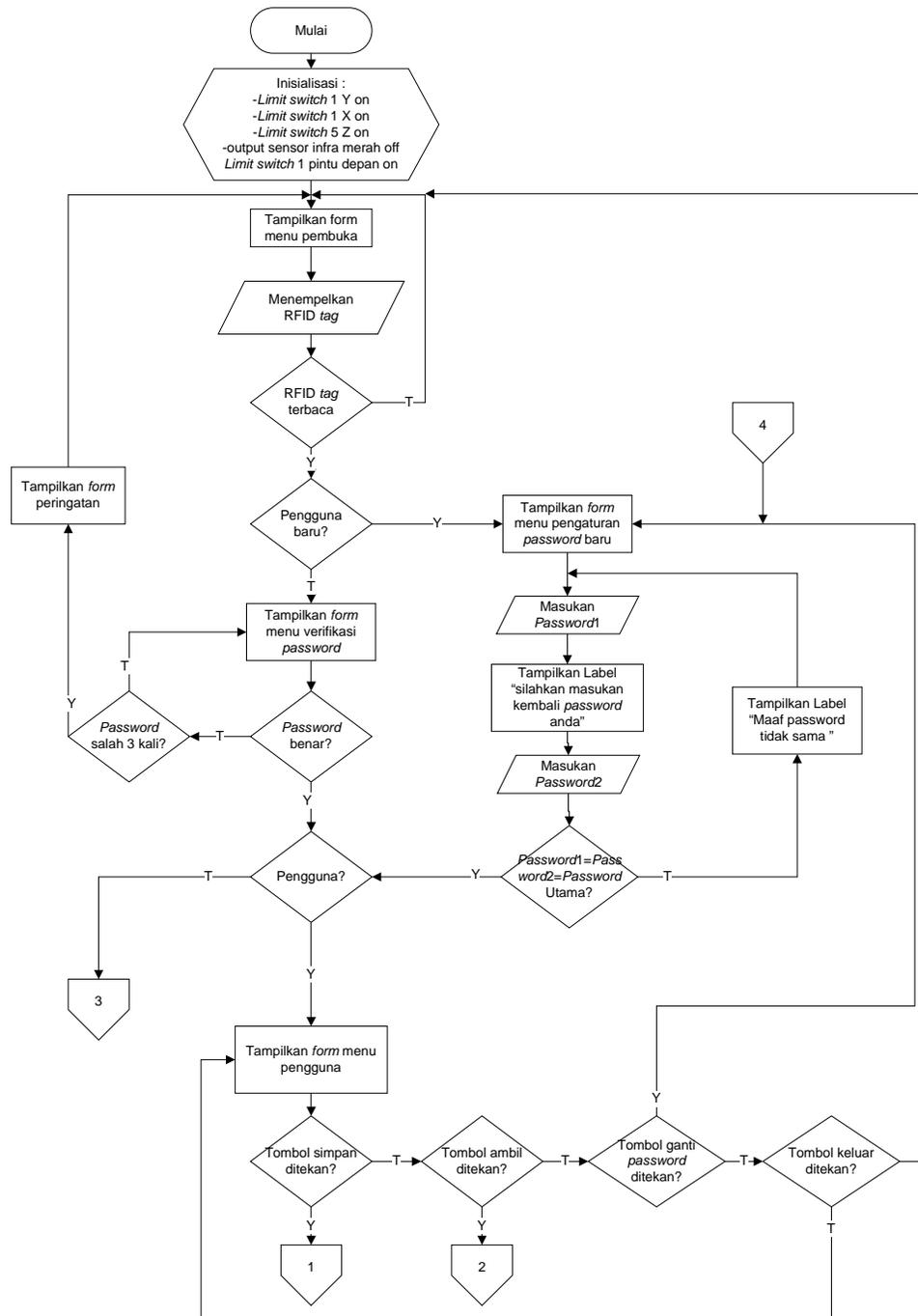
bisa mengakses *slot* tersebut hanya pengguna yang sebelumnya memilih.

- g. Tombol menyimpan benda akan menjadi tidak bisa ditekan ketika semua *slot* sudah terisi.

3) Pengambilan Benda

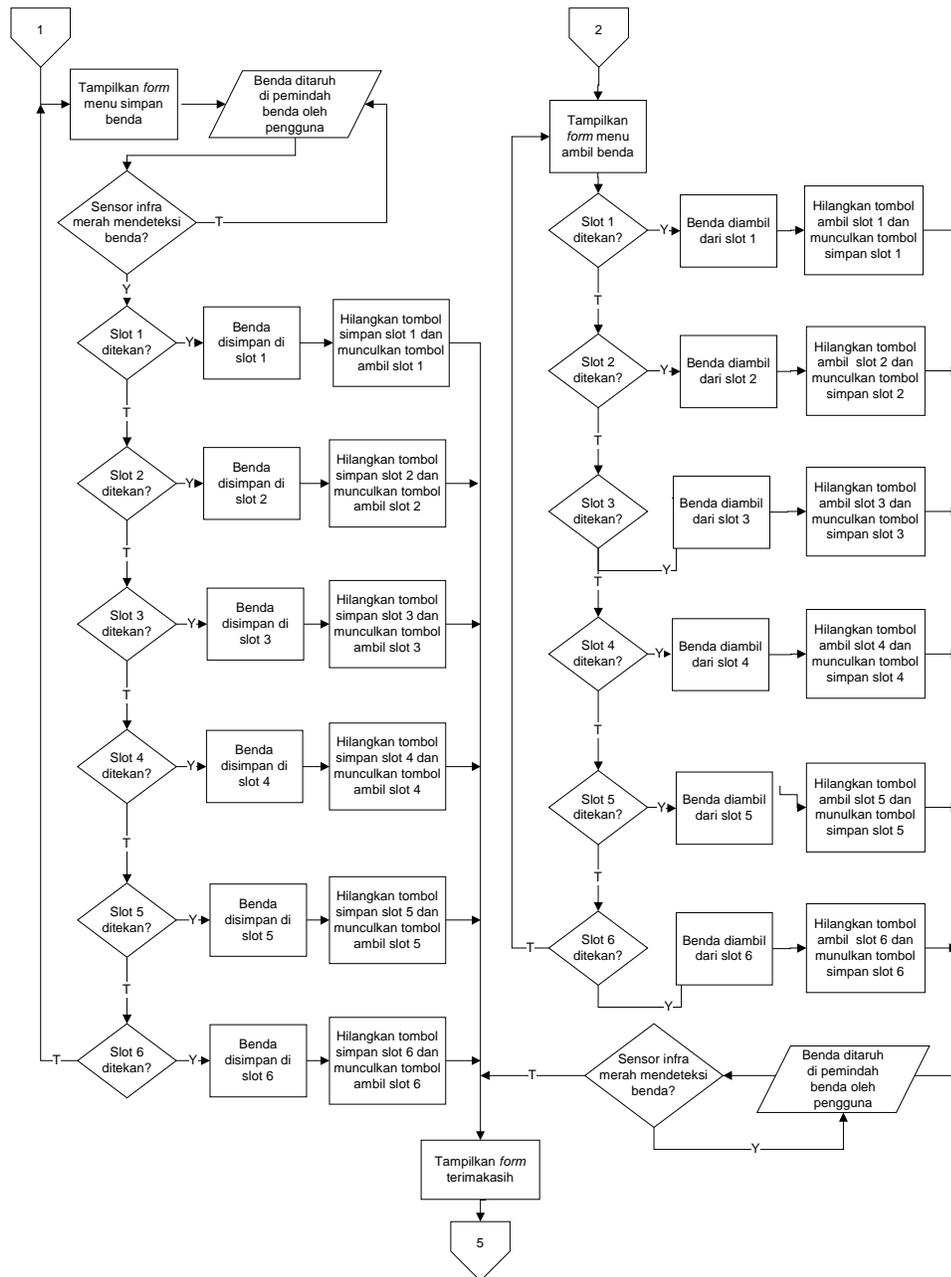
- a. Pada *form* menu pengguna tekan tombol ambil untuk mengambil benda. Jika pengguna belum pernah menyimpan benda maka tombol ambil tidak dapat ditekan oleh pengguna.
- b. Menu *form* pengambilan benda muncul pada layar, dimana pada tiap tombol *slot* terdapat data tentang kapan pengguna menyimpan benda di *slot* tersebut.
- c. Pengguna memilih *slot* mana yang bendanya akan diambil.
- d. Setelah memilih *slot*, data penyimpanan di *slot* tersebut akan dihapus dan alat akan mengambil benda pada *slot* tersebut. Ketika benda sudah sampai di posisi sirkulasi pintu alat penyimpanan akan terbuka.
- e. Ketika benda sudah diambil maka pintu akan kembali menutup dan *slot* kosong tersebut kembali bisa diakses oleh semua pengguna.

Untuk lebih jelasnya sistem yang bersifat umum tersebut disusun dalam *flowchart* seperti yang ditunjukkan oleh gambar 3.10.

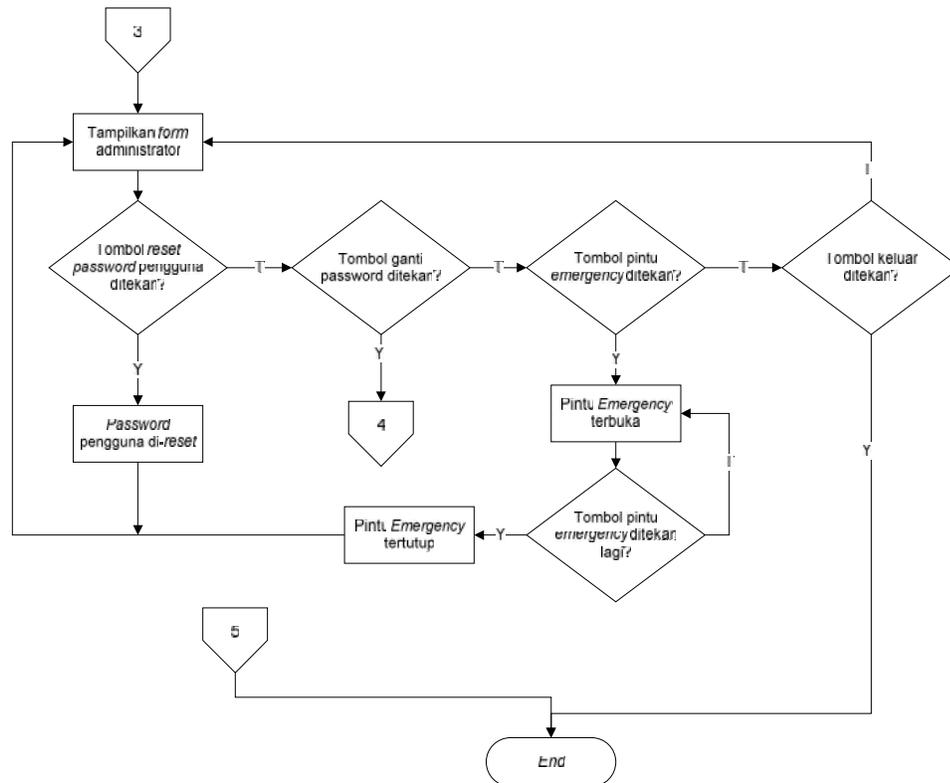


Gambar 3.10. (a) Flowchart *Prototype* Penyimpanan Benda Otomatis

Sumber : Dokumen Pribadi



Gambar 3. 10. (b) Flowchart Prototype Penyimpanan Benda Otomatis
 Sumber : Dokumen Pribadi



Gambar 3.10. (c) Flowchart Prototype Penyimpanan Benda Otomatis
Sumber : Dokumen Pribadi

3.3.4.2. Penetapan Alamat *Input* dan Alamat *Output* PLC

PLC yang digunakan untuk mengendalikan *input* dan *output* dalam sistem kerja *prototype* penyimpanan benda bertingkat otomatis merupakan PLC Omron CJ1M. Di mana pada alat ini menggunakan 13 *input* dan 15 *input* dari *visual basic*, serta 9 *output* digital. Untuk lebih jelasnya dapat dilihat pada tabel 3.1, tabel 3.2, dan tabel 3.3.

Tabel 3.1. Alamat *Input Digital PLC*

No.	<i>Input PLC</i>	Alamat	Keterangan
1.	<i>Limit switch</i> 1 motor X	0.00	Berfungsi untuk membatasi pergerakan maju pemindah benda.
2.	<i>Limit switch</i> 2 motor X	0.01	Berfungsi untuk membatasi pergerakan mundur pemindah benda.
3.	<i>Limit switch</i> 1 motor Y	0.02	Berfungsi untuk membatasi pergerakan ke kanan pemindah benda.
4.	<i>Limit switch</i> 2 motor Y	0.03	Berfungsi untuk membatasi pergerakan ke kiri pemindah benda.
5.	<i>Limit switch</i> 1 motor Z	0.04	Berfungsi untuk membatasi pergerakan pemindah benda naik ke titik lebih rendah untuk mengambil benda pada tingkat 1.
6.	<i>Limit switch</i> 2 motor Z	0.05	Berfungsi untuk membatasi pergerakan pemindah benda naik ke titik lebih tinggi untuk menaruh benda pada tingkat 1
7.	<i>Limit switch</i> 3 motor Z	0.06	Berfungsi untuk membatasi pergerakan pemindah benda naik ke titik lebih rendah untuk mengambil benda pada tingkat 2.
8.	<i>Limit switch</i> 4 motor Z	0.07	Berfungsi untuk membatasi pergerakan pemindah benda naik ke titik lebih tinggi untuk menaruh benda pada tingkat 2.
9.	<i>Limit switch</i> 5 motor Z	0.08	Berfungsi untuk membatasi pergerakan pemindah benda naik ke titik lebih rendah untuk mengambil benda pada tingkat 3.
10.	<i>Limit switch</i> 6 motor Z	0.09	Berfungsi untuk membatasi pergerakan pemindah benda naik ke titik lebih tinggi untuk menaruh benda pada tingkat 3.
11.	<i>Limit Switch</i> 1 Motor P	0.10	Berfungsi untuk membatasi pergerakan pintu menutup.
12.	<i>Limit Switch</i> 2 Motor P	0.11	Berfungsi untuk membatasi pergerakan pintu membuka.
13.	Sensor infra merah	0.12	Berfungsi untuk mendeteksi benda di pemindah benda.

Sumber : Dokumen Pribadi

Tabel 3.2. Alamat Output Digital PLC

No.	Output PLC	Alamat	Keterangan
1.	Relai daya motor X	1.00	Berfungsi sebagai penghubung dan pemutus daya pada motor X
2.	Relai arah motor X	1.01	Berfungsi sebagai perubah arah ke depan atau belakang pada motor X
3.	Relai daya motor Y	1.02	Berfungsi sebagai penghubung dan pemutus daya pada motor Y
4.	Relai arah motor Y	1.03	Berfungsi sebagai perubah arah ke kiri atau kanan pada motor Y
5.	Relai daya motor Z	1.04	Berfungsi sebagai penghubung dan pemutus daya pada motor Z
6.	Relai arah motor Z	1.05	Berfungsi sebagai perubah arah ke atas atau bawah pada motor Z
7.	Relai daya motor pintu	1.06	Berfungsi sebagai penghubung dan pemutus daya pada motor pintu
8.	Relai arah motor pintu	1.07	Berfungsi sebagai perubah arah ke kiri atau kanan pada motor pintu
9.	Relai daya <i>doorlock</i>	1.08	Berfungsi sebagai penghubung dan pemutus daya pada <i>doorlock</i>

Sumber : Dokumen Pribadi

Tabel 3.3. Alamat Visual Basic PLC

No.	Input Visual Basic	Alamat	Keterangan
1.	Tombol simpan	A448.00	Berfungsi untuk memilih fungsi simpan
2.	Tombol ambil	A448.01	Berfungsi untuk memilih fungsi ambil
3.	Tombol slot 1 simpan	A448.02	Berfungsi sebagai perintah menyimpan di <i>slot 1</i>
4.	Tombol slot 2 simpan	A448.03	Berfungsi sebagai perintah menyimpan di <i>slot 2</i>
5.	Tombol slot 3 simpan	A448.04	Berfungsi sebagai perintah menyimpan di <i>slot 3</i>
6.	Tombol slot 4 simpan	A448.05	Berfungsi sebagai perintah menyimpan di <i>slot 4</i>
7.	Tombol slot 5 simpan	A448.06	Berfungsi sebagai perintah menyimpan di <i>slot 5</i>
8.	Tombol slot 6 simpan	A448.07	Berfungsi sebagai perintah menyimpan di <i>slot 6</i>
9.	Tombol slot 1 ambil	A448.09	Berfungsi sebagai perintah mengambil dari <i>slot 1</i>

No.	<i>Input Visual Basic</i>	Alamat	Keterangan
10.	Tombol slot 2 ambil	A448.10	Berfungsi sebagai perintah mengambil dari <i>slot 2</i>
11.	Tombol slot 3 ambil	A448.11	Berfungsi sebagai perintah mengambil dari <i>slot 3</i>
12.	Tombol slot 4 ambil	A448.12	Berfungsi sebagai perintah mengambil dari <i>slot 4</i>
13.	Tombol slot 5 ambil	A448.13	Berfungsi sebagai perintah mengambil dari <i>slot 5</i>
14.	Tombol slot 6 ambil	A448.14	Berfungsi sebagai perintah mengambil dari <i>slot 6</i>
15.	Tombol <i>emergency</i>	A449.00	Berfungsi untuk membuka pintu darurat pada rak

Sumber : Dokumen Pribadi

3.3.4.3. Program PLC

Sistem pengendalian *prototype* penyimpanan benda bertingkat otomatis pada PLC menggunakan bahasa pemrograman diagram *ladder*. Pembuatan diagram *ladder* merupakan langkah lanjutan dalam perancangan program PLC setelah menetapkan rancangan alat dan alamat *input* serta *output* PLC-nya. Diagram *ladder* dalam sistem *prototype* penyimpanan benda bertingkat otomatis dapat dilihat pada lembar lampiran.

3.3.4.4. Program *visual basic*

Setelah program PLC sudah ditentukan maka tahap selanjutnya menentukan program *visual basic*. Dimana pada *visual basic* sistem pengaman alat dirancang dan dibuat. *Software Visual basic* juga merupakan bagian yang menghubungkan antara PLC dan RFID. Pada

tahap ini kita menentukan tampilan *interface* alat dengan pengguna. Untuk tampilan dan *coding* dari *software* yang dibuat di *visual basic* dapat dilihat pada lembar lampiran.

3.4. Instrumen Penelitian

Instrumen pada penelitian merupakan lembar pengamatan alat yang berisi kolom untuk mencatat hasil pengujian alat yang terdiri dari beberapa pengamatan.

Adapun alat bantu penelitian yang digunakan pada penelitian yaitu :

1. Multitester Heles YX-393, digunakan untuk mengukur tegangan, arus dan mengecek sambungan kabel pada rangkaian alat penelitian.
2. Nampan slot, digunakan untuk menguji kerja sensor infra merah.
3. PC (*Personal Computer*) / *Notebook*, digunakan untuk menulis *ladder diagram*, simulasi, kompilasi, transfer program ke PLC, membuat *software*, serta koneksi RFID dengan alat.
4. PLC CJ1M, digunakan untuk mengetahui *input* dan *output* dari program yang telah dibuat.
5. *Software* CX-Programmer, digunakan untuk pembuatan dan uji coba program *ladder diagram* pada PLC.
6. *Software visual basic 6.0*, digunakan untuk pembuatan dan uji coba program *interface* antara pengguna dan alat.
7. *Stopwatch*, digunakan untuk menguji waktu kerja penyimpanan dan pengambilan benda.

Kriteria pengujian alat yang dilakukan pada penilitan adalah dengan menggunakan metode pengujian secara langsung (percobaan), sehingga diketahui apakah peralatan yang akan dikendalikan sesuai dengan program yang telah dirancang. Pengujian alat dilaksanakan ketika alat telah jadi dan siap digunakan. Pengujian alat akan dilakukan dengan mencoba menyimpan dan mengambil benda.

3.4.1. Pengujian Sensor Infra Merah

Rangkaian sensor infra merah digunakan untuk memberikan sinyal apakah pada slot tersebut sudah terdapat benda atau belum. Komponen rangkaian yang akan diuji adalah tegangan *input* dan *output*, photodioda serta transistor C828. Pengujian dilakukan dengan cara mengukur besarnya tegangan. Di mana kriteria pengujian sensor infra merah dapat dilihat pada tabel 3.4.

Tabel 3.4. Pengujian Sensor Infra Merah

V_{in} (Volt)	Keadaan	V basis C828 (Volt)	V_{out} (Volt)	Kondisi
.....	Terhalang
.....	Tidak terhalang

Sumber : Dokumen Pribadi

3.4.2. Pengujian RFID

Pada alat ini RFID memegang peranan yang sangat penting, yaitu sebagai akses utama untuk menggunakan alat. Dimana pada tiap-tiap RFID *tag* memiliki ID yang berbeda yang menentukan identitas pengguna pada

software. Pengujian dilakukan dengan cara mengukur jarak pembacaan RFID *reader* terhadap RFID *tag*. Di mana kriteria pengujian sensor infra merah dapat dilihat pada tabel 3.5.

Tabel 3.5. Pengujian RFID

Vin	RFID tag		Jarak Baca (cm)	Kondisi
	Nomor Kartu	ID		
.....	1	5
.....	1	4
.....	1	3
.....	1	2
.....	1	1
.....	2	5
.....	2	4
.....	2	3
.....	2	2
.....	2	1
.....	3	5
.....	3	4
.....	3	3
.....	3	2
.....	3	1

Sumber : Dokumen Pribadi

3.4.3. Pengujian tegangan *Prototype*

Pengujian sistem kerja *prototype* secara keseluruhan dilakukan setelah serangkaian pengujian sebelumnya berjalan dengan lancar. Pengujian ini bertujuan untuk melihat apakah terdapat tegangan pada setiap *input* dan

output prototype ini, dan setiap komponen dapat bekerja dengan normal.

Dimana kriteria pengujian dapat dilihat pada tabel 3.6.

Tabel 3.6. Pengujian Tegangan pada *Prototype*

No.	Jenis Item	Alamat	Tegangan Pada Soket PLC		Tegangan Pada Coil Relai	
			<i>Off</i>	<i>On</i>	<i>Off</i>	<i>On</i>
1.	<i>Limit Switch 1 X</i>	0.00
2.	<i>Limit Switch 2 X</i>	0.01
3.	<i>Limit Switch 1 Y</i>	0.02
4.	<i>Limit Switch 2 Y</i>	0.03
5.	<i>Limit Switch 1 Z</i>	0.04
6.	<i>Limit Switch 2 Z</i>	0.05
7.	<i>Limit Switch 3 Z</i>	0.06
8.	<i>Limit Switch 4 Z</i>	0.07
9.	<i>Limit Switch 5 Z</i>	0.08
10.	<i>Limit Switch 6 Z</i>	0.09
11.	<i>Limit Switch 1 P</i>	0.10
12.	<i>Limit Switch 2 P</i>	0.11
13.	Sensor Infra merah	0.12
14.	Relai Daya Motor X	1.00
15.	Relai Arah Motor X	1.01
16.	Relai Daya Motor Y	1.02
17.	Relai Arah Motor Y	1.03
18.	Relai Daya Motor Z	1.04
19.	Relai Arah Motor Z	1.05
20.	Relai Daya Motor P	1.06
21.	Relai Arah Motor P	1.07
22.	Relai Doorlock	1.08

Sumber : Dokumen Pribadi

3.4.4. Pengujian Sistem Pengaman

Pengujian sistem pengaman pada alat ini berguna untuk memastikan apakah pengaman RFID dan password dapat bekerja dengan seharusnya. Kriteria pengujian sistem pengaman dapat dilihat pada tabel 3.7.

Tabel 3.7. Pengujian Sistem Pengaman

RFID tag	<i>Password Default</i>	<i>Password Log in</i>	Setelah tiga kali	Kondisi
.....
.....
.....
.....
.....
.....
.....

Sumber : Dokumen Pribadi

3.4.5. Pengujian Penyimpanan dan Pengambilan *Software Visual Basic*

Agar mendapatkan kepastian pada saat penyimpanan dan pengambilan pada alat ini maka pengujian penyimpanan dan pengambilan pada *software visual basic* harus dilakukan untuk memastikan benda disimpan pada slot yang sesuai dengan yang dipilih. Ketika tombol ditekan kemudian melihat hasil tampilan pada menu pengguna baik itu menu pengguna 1 atau pun 2. Adapun

kriteria pengujian penyimpanan dan pengambilan *software visual basic* dapat dilihat pada tabel 3.8 dan 3.9.

Tabel 3.8. Pengujian Tombol Penyimpanan

Pengguna	Tombol Simpan yang ditekan	Tampilan tombol simpan pengguna 1	Tampilan tombol simpan pengguna 2	Tampilan tombol ambil pengguna 1	Tampilan tombol ambil pengguna 2	Ket.
Pengguna 1	Slot 1
Pengguna 1	Slot 2
Pengguna 2	Slot 6
Pengguna 2	Slot 5
Pengguna 1	Slot 3
Pengguna 2	Slot 4

Sumber : Dokumen Pribadi

Tabel 3.9. Pengujian Tombol Pengambilan

Pengguna	Tombol Ambil yang ditekan	Tampilan tombol simpan pengguna 1	Tampilan tombol simpan pengguna 2	Tampilan tombol ambil pengguna 1	Tampilan tombol ambil pengguna 2	Ket.
Pengguna 1	Slot 1
Pengguna 1	Slot 2
Pengguna 2	Slot 6
Pengguna 2	Slot 5
Pengguna 1	Slot 3
Pengguna 2	Slot 4

Sumber : Dokumen Pribadi

3.4.6. Pengujian Waktu Kerja

Pengujian waktu kerja dilakukan guna mengetahui apakah *prototype* dapat bekerja dengan waktu yang efektif dari titik awal ke tiap slot penyimpanan.

Tabel pengujian waktu kerja dapat dilihat pada tabel 3.10 dan 3.11.

Tabel 3.10. Pengujian Waktu Tempuh Penyimpanan

No.	Slot yang Dituju	Waktu Tempuh	Keterangan
1.	Slot 1
2.	Slot 2
3.	Slot 3
4.	Slot 4
5.	Slot 5
6.	Slot 6

Sumber : Dokumen Pribadi

Tabel 3.11. Pengujian Waktu Tempuh Pengambilan

No.	Asal Slot	Waktu Tempuh	Keterangan
1.	Slot 1
2.	Slot 2
3.	Slot 3
4.	Slot 4
5.	Slot 5
6.	Slot 6

Sumber : Dokumen Pribadi

BAB IV

HASIL PENELITIAN

4.1. Hasil Pengujian Alat

4.1.1. Hasil Pengujian Sensor Infra Merah

Pengujian sensor infra merah dilakukan dengan cara menghalangi antara fotodiode dan led infra merah dengan nampan, kemudian melihat kondisi sinyal outputnya. Selain sinyal output tegangan input dan tegangan output yang menuju relai sensor pun menjadi titik pengujian. Hasil pengujian sensor infra merah dapat dilihat pada tabel 4.1.

Tabel 4.1. Hasil Pengujian Sensor Infra Merah

Vin (Volt)	Keadaan	V basis C828 (Volt)	Vout (Volt)	Kondisi Sinyal
5.9	Terhalang	0.2	0.075	<i>On</i>
5.9	Tidak terhalang	0.8	5.8	<i>Off</i>

Sumber : Dokumen Pribadi

4.1.2. Hasil Pengujian RFID

Pengujian ini dilakukan dengan cara mendekatkan RFID *tag* terhadap RFID *reader* dengan jarak tertentu. Sehingga kita bisa mendapatkan informasi mengenai maksimal jarak pembacaan. Hasil pengujian RFID dapat dilihat pada tabel 4.2.

Tabel 4.2. Hasil Pengujian RFID

Vin (Volt)	RFID tag		Jarak Baca (cm)	Kondisi
	Nomor Kartu	ID		
11,5	1	-	5	Tidak terbaca
11,5	1	0700A184E2C0	4	Terbaca
11,5	1	0700A184E2C0	3	Terbaca
11,5	1	0700A184E2C0	2	Terbaca
11,5	1	0700A184E2C0	1	Terbaca
11,5	2	-	5	Tidak terbaca
11,5	2	0700A10717B6	4	Terbaca
11,5	2	0700A10717B6	3	Terbaca
11,5	2	0700A10717B6	2	Terbaca
11,5	2	0700A10717B6	1	Terbaca
11,5	3	-	5	Tidak terbaca
11,5	3	0700A11915AA	4	Terbaca
11,5	3	0700A11915AA	3	Terbaca
11,5	3	0700A11915AA	2	Terbaca
11,5	3	0700A11915AA	1	Terbaca

Sumber : Dokumen Pribadi

4.1.3. Hasil Pengujian Tegangan *Prototype*

Hasil pengujian tegangan *prototype* disini merupakan hasil pengujian tegangan masukan pada saat input aktif atau tidak. Pengujian ini dilakukan dengan mengukur tegangan pada soket PLC dan kumparan dari masing-masing relai.

Pengujian menggunakan *multitester* yang diatur pada posisi volt tegangan searah (DC). Pengujian tegangan soket PLC pada *limit switch* dan sensor infra

merah dilakukan dengan menaruh *probe* negatif (-) di COM input PLC dan *probe* positif (+) di soket masing-masing *address input* PLC. Untuk pengujian pada *coil* relai *output* yang yang digunakan untuk relai motor dan *doorlock* dengan menaruh langsung *probe* positif dan negatif *multitester* di *coil* relai. Sedangkan pada pengujian *coil output* yang ada pada PLC dengan menaruh pada soket masing-masing *address output* PLC dengan *probe* negatif dan *probe* positif di COM *output* PLC. Hasil pengujian tegangan *prototype* dapat dilihat pada tabel 4.3.

Tabel 4.3. Hasil Pengujian Tegangan *Prototype*

No.	Jenis Item	Alamat	Tegangan Pada Soket PLC (Volt)		Tegangan Pada Coil Relai (Volt)	
			<i>Off</i>	<i>On</i>	<i>Off</i>	<i>On</i>
1.	<i>Limit Switch</i> 1 X	0.00	0	23.5	-	-
2.	<i>Limit Switch</i> 2 X	0.01	0	23.5	-	-
3.	<i>Limit Switch</i> 1 Y	0.02	0	23.5	-	-
4.	<i>Limit Switch</i> 2 Y	0.03	0	23.5	-	-
5.	<i>Limit Switch</i> 1 Z	0.04	0	23.5	-	-
6.	<i>Limit Switch</i> 2 Z	0.05	0	23.5	-	-
7.	<i>Limit Switch</i> 3 Z	0.06	0	23.5	-	-
8.	<i>Limit Switch</i> 4 Z	0.07	0	23.5	-	-
9.	<i>Limit Switch</i> 5 Z	0.08	0	23.5	-	-
10.	<i>Limit Switch</i> 6 Z	0.09	0	23.5	-	-
11.	<i>Limit Switch</i> 1 P	0.10	0	23.5	-	-
12.	<i>Limit Switch</i> 2 P	0.11	0	23.5	-	-
13.	Sensor Infra merah	0.12	0	23.5	-	-
14.	Relai Daya Motor X	1.00	23.5	0	0	23.5
15.	Relai Arah Motor X	1.01	23.5	0	0	23.5

No.	Jenis Item	Alamat	Tegangan Pada Soket PLC (Volt)		Tegangan Pada Coil Relai (Volt)	
			<i>Off</i>	<i>On</i>	<i>Off</i>	<i>On</i>
16.	Relai Daya Motor Y	1.02	23.5	0	0	23.5
17.	Relai Arah Motor Y	1.03	23.5	0	0	23.5
18.	Relai Daya Motor Z	1.04	23.5	0	0	23.5
19.	Relai Arah Motor Z	1.05	23.5	0	0	23.5
20.	Relai Daya Motor P	1.06	23.5	0	0	23.5
21.	Relai Arah Motor P	1.07	23.5	0	0	23.5
22.	Relai Doorlock	1.08	23.5	0	0	23.5

Sumber : Dokumen Pribadi

4.1.4. Hasil Pengujian Sistem Pengaman

Tahap pertama dalam pengujian adalah menentukan *password default* pada tiap-tiap kartu terlebih dahulu. Kemudian mencoba masuk dengan *password* yang sesuai dan tidak sesuai. Hasil pengujian sistem pengaman dapat dilihat pada tabel 4.4.

Tabel 4.4. Hasil Pengujian Sistem Pengaman

RFID tag	<i>Password Default</i>	<i>Password Log in</i>	Setelah tiga kali	Kondisi
0700A184E2C0	111111	111111	-	Berhasil <i>log in</i>
0700A184E2C0	111111	222222	Muncul <i>form</i> peringatan	Tidak berhasil
0700A184E2C0	111111	333333	Muncul <i>form</i> peringatan	Tidak berhasil
0700A10717B6	222222	111111	Muncul <i>form</i> peringatan	Tidak berhasil
0700A10717B6	222222	222222	-	Berhasil <i>log in</i>
0700A10717B6	222222	333333	Muncul <i>form</i> peringatan	Tidak berhasil

RFID tag	<i>Password Default</i>	<i>Password Log in</i>	Setelah tiga kali	Kondisi
0700A11915AA	333333	111111	Muncul <i>form</i> peringatan	Tidak berhasil
0700A11915AA	333333	222222	Muncul <i>form</i> peringatan	Tidak berhasil
0700A11915AA	333333	333333	-	Berhasil <i>log in</i>

Sumber : Dokumen Pribadi

4.1.5. Hasil Pengujian Penyimpanan dan Pengambilan *Software Visual*

Basic

Ketika sistem pengaman sudah berjalan dengan baik maka proses penyimpanan dan pengambilan pada *software* dilakukan. Pengujian ini dilakukan dengan cara menyimpan dan mengambil benda dengan 2 ID yang berbeda, sehingga kita dapat memastikan apakah ketika kita menyimpan benda dapat diakses oleh pengguna lain atau tidak. Hasil pengujian penyimpanan dan pengambilan *software visual basic* dapat dilihat pada tabel 4.5 dan 4.6.

Tabel 4.5. Hasil Pengujian Penyimpanan *Software*

Pengguna	Tombol Simpan yang ditekan	Tampilan tombol simpan pengguna 1	Tampilan tombol simpan pengguna 2	Tampilan tombol ambil pengguna 1	Tampilan tombol ambil pengguna 2	Ket.
Pengguna 1	Slot 1	Slot 2, 3, 4, 5, 6	Slot 2, 3, 4, 5, 6	Slot 1	Tidak dapat diakses	Sesuai
Pengguna 1	Slot 2	Slot 3, 4, 5, 6	Slot 3, 4, 5, 6	Slot 1, 2	Tidak dapat diakses	Sesuai

Pengguna	Tombol Simpan yang ditekan	Tampilan tombol simpan pengguna 1	Tampilan tombol simpan pengguna 2	Tampilan tombol ambil pengguna 1	Tampilan tombol ambil pengguna 2	Ket.
Pengguna 2	Slot 6	Slot 3, 4, 5	Slot 3, 4, 5	Slot 1, 2	Slot 6	Sesuai
Pengguna 2	Slot 5	Slot 3, 4	Slot 3, 4	Slot 1, 2	Slot 5, 6	Sesuai
Pengguna 1	Slot 3	Slot 4	Slot 4	Slot 1, 2, 3	Slot 5, 6	Sesuai
Pengguna 2	Slot 4	Tidak dapat diakses	Tidak dapat diakses	Slot 1, 2, 3	Slot 4, 5, 6	Sesuai

Sumber : Dokumen Pribadi

Tabel 4.6. Hasil Pengujian Pengambilan *Software*

Pengguna	Tombol Ambil yang ditekan	Tampilan tombol simpan pengguna 1	Tampilan tombol simpan pengguna 2	Tampilan tombol ambil pengguna 1	Tampilan tombol ambil pengguna 2	Ket.
Pengguna 1	Slot 1	Slot 1	Slot 1	Slot 2, 3	Slot 4, 5, 6	Sesuai
Pengguna 1	Slot 2	Slot 1, 2	Slot 1, 2	Slot 3	Slot 4, 5, 6	Sesuai
Pengguna 2	Slot 6	Slot 1, 2, 6	Slot 1, 2, 6	Slot 3	Slot 4, 5	Sesuai
Pengguna 2	Slot 5	Slot 1, 2, 5, 6	Slot 1, 2, 5, 6	Slot 3	Slot 4	Sesuai
Pengguna 1	Slot 3	Slot 1, 2, 3, 5, 6	Slot 1, 2, 3, 5, 6	Tidak dapat diakses	Slot 4	Sesuai
Pengguna 2	Slot 4	Slot 1, 2, 3, 4, 5, 6	Slot 1, 2, 3, 4, 5, 6	Tidak dapat diakses	Tidak dapat diakses	Sesuai

Sumber : Dokumen Pribadi

4.1.6. Hasil Pengujian Waktu Kerja

Setelah *prototype* berhasil dibuat, pengujian waktu kerja baru bisa dilakukan dengan cara menguji untuk menyimpan dan mengambil benda pada tiap-tiap slot. Hasil dari pengujian waktu kerja dapat dilihat pada tabel 4.5 dan 4.6.

Tabel 4.7. Hasil Pengujian Waktu Kerja Penyimpanan

No.	Slot yang Dituju	Waktu Tempuh	Keterangan
1.	Slot 1	48 detik	Sesuai dengan pilihan
2.	Slot 2	53 detik	Sesuai dengan pilihan
3.	Slot 3	38 detik	Sesuai dengan pilihan
4.	Slot 4	46 detik	Sesuai dengan pilihan
5.	Slot 5	31 detik	Sesuai dengan pilihan
6.	Slot 6	46 detik	Sesuai dengan pilihan

Sumber : Dokumen Pribadi

Tabel 4.8. Hasil Pengujian Waktu Kerja Pengambilan

No.	Asal Slot	Waktu Tempuh	Keterangan
1.	Slot 1	46 detik	Sesuai dengan pilihan
2.	Slot 2	53 detik	Sesuai dengan pilihan
3.	Slot 3	38 detik	Sesuai dengan pilihan
4.	Slot 4	48 detik	Sesuai dengan pilihan
5.	Slot 5	31 detik	Sesuai dengan pilihan
6.	Slot 6	44 detik	Sesuai dengan pilihan

Sumber : Dokumen Pribadi

4.2. Analisis Hasil Pengujian Alat

4.2.1. Analisis Hasil Pengujian Sensor Infra Merah

Pada tegangan 6 VDC infra merah akan memancarkan sinarnya yang kemudian diterima oleh fotodioda, kemudian akan diolah oleh rangkaian penerima agar menghasilkan data *biner*. Jika fotodioda menerima pancaran sinar infra merah maka *output* dari rangkaian penerima ini akan mengeluarkan logika *low* (0), namun jika fotodioda tidak menerima pantulan sinar infra merah, maka *output* dari rangkaian penerima akan mengeluarkan logika *high* (1) hal ini karena rangkaian dihubungkan dengan kaki NC relai.

Seperti yang sudah dijelaskan pada bab 3 mengenai perancangan rangkaian sensor infra merah, fotodioda memiliki hambatan sekitar 15 s/d 20 MOhm jika tidak terkena sinar infra merah, dan hambatannya akan berubah menjadi sekitar 80 s/d 300 KOhm jika terkena sinar infra merah tergantung dari besarnya intensitas yang mengenainya. Semakin besar intensitasnya, maka hambatannya semakin kecil.

Kemudian *output* dari photodioda diumpankan ke basis dari transistor tipe NPN C828, dimana untuk membuat transistor tersebut aktif maka tegangan yang keluar dari photodioda lebih besar dari 0,7 VDC, syarat ini terpenuhi ketika fotodioda mendapat sinar infra merah, karena ketika ada sinar infra merah yang mengenai fotodioda, maka hambatan pada fotodioda 300Kohm. Secara teoritis analisisnya sebagai berikut:

$$V_o = \frac{R_2}{R_1+R_2} \times V_c = \frac{4}{3+4} \times 6 = 0.79 V \quad \dots\dots (4.1)$$

V_{out} akan diumpankan ke basis dari transistor C828, karena tegangannya lebih besar dari 0,7 VDC yaitu 0.79 VDC maka transistor akan aktif, begitu pula hasil penelitian pada sensor infra merah yang digunakan pada alat ini, yang tertera pada tabel 4.1, V_{basic} transistor C828 memiliki tegangan sebesar 0.8 VDC.

Sedangkan jika ada benda yang menghalangi sensor, maka pancaran infra merah yang mengenai photodiode akan terhalang. Hal ini menyebabkan hambatan pada photodiode berubah dari 300 KOhm menjadi 15 MOhm, sehingga jika V_{out} yg diumpankan ke basis transistor C828 tegangannya hanya 0,02 VDC maka transistor tidak aktif. Tidak aktifnya transistor C828 akan menyebabkan kolektornya tidak terhubung ke emitor, sehingga kolektor tidak mendapat tegangan 6 VDC dari V_{cc} .

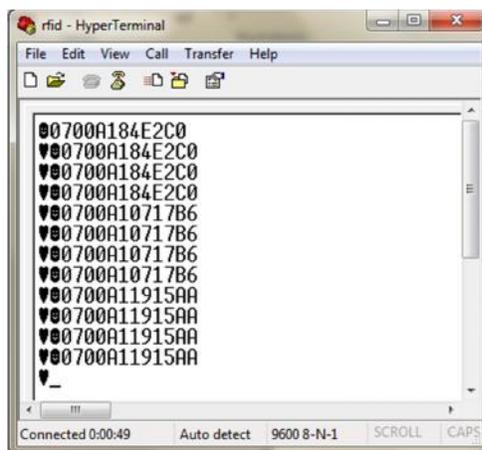
$$V_o = \frac{R_2}{R_1+R_2} \times V_c = \frac{4}{1+4} \times 6 = 0.02 V \quad \dots\dots\dots(4.2)$$

4.2.2. Analisis Hasil Pengujian RFID

Pengujian dilakukan dengan cara memberi RFID *reader* ID-12 tegangan *input* (V_{in}) sebesar 11,5 VDC sehingga RFID *reader* dapat bekerja. Setelah itu RFID *reader* dihubungkan dengan *software hyperterminal* guna mengetahui ID yang terdapat pada masing-masing RFID *tag*. Dari hasil

percobaan tampak jelas ketika RFID *reader* diberikan tegangan sebesar 11,5 VDC jarak maksimum yang dapat dibaca oleh RFID *reader* adalah sebesar 4 cm, terhitung dari permukaan ID-12 sampai permukaan RFID *tag* menggunakan mistar. Walaupun pada *datasheet* IC ID-12 tercantum bahwa jarak pembacaan bisa sampai 12+cm.

Berdasarkan tabel 4.2 Tiap-tiap RFID *tag* memiliki nomor ID sebanyak 12 digit, dimana 12 digit tersebut terdiri dari beberapa huruf dan angka yang pada tiap RFID *tag* memiliki 12 digit yang berbeda-beda. Pada kartu 1 memiliki ID 0700A184E2C0, kartu 2 memiliki ID 0700A10717B6, dan kartu 3 memiliki ID 0700A11915AA. Setelah melakukan pengujian jarak baca dan nomor ID ini, kemudian ID yang terbaca tersebut dimasukan kedalam *software* yang dibuat di *visual basic* untuk memberikan akses pada *software* pada masing-masing ID tersebut untuk menggunakan penyimpanan otomatis. Tampilan hyperterminal saat pengujian ditunjukkan oleh gambar 4.1.



Gambar 4. 1. Pengujian Melalui *Hyperterminal*
Sumber : Dokumen Pribadi

4.2.3. Analisis Hasil Pengujian Tegangan *Prototype*

Pada *prototype* ini digunakan 12 buah *limit switch* dan satu buah sensor infra merah dan 9 relai. Berdasarkan tabel 4.3, tegangan *limit switch* ketika tidak terjadi kontak pada soket PLC adalah 0 VDC, dan pada saat terjadi kontak sebesar 23,5 VDC. Hal tersebut sama nilainya seperti pada sinyal yang diberikan oleh sensor infra merah pada soket PLC, yaitu 0 VDC ketika tidak ada benda yang menghalangi dan 23,5 VDC ketika ada benda yang menghalangi.

Sedangkan tegangan pada tiap-tiap soket *output* yang terhubung dengan relai memiliki nilai 23,5 VDC. Namun pada soket tiap-tiap relai memiliki nilai 0 VDC. Hal ini karena tegangan +VDC dari *power supply* terhubung dengan soket COM pada *output* PLC. Jadi ketika *output* PLC *high* (1) tegangan pada soket PLC menjadi 0VDC dan pada soket relai menjadi 23,5VDC.

4.2.4. Analisis Hasil Pengujian Sistem Pengaman

Pada saat pengujian sistem pengaman berjalan dengan sesuai. Dimana ketika RFID *tag* dan *password* tidak bersesuaian satu sama lain maka pengguna tidak dapat *log in*. Setelah tiga kali pengguna tidak berhasil *log in* maka akan muncul *form* peringatan. Dan mengembalikan tampilan kembali pada *form* awal. Adapun rangkaian *coding* yang mengatur agar hal tersebut berjalan adalah sebagai berikut :

```

Public passn, passy, passa As Single
Public error As Integer
Public nilaiinisial As Integer
Public user1, user2, user3 As Single

Private Sub Timer1_Timer()
RS = Text1.Text
ID = Mid(RS, 2, 12)
If ID = "0700A184E2C0" And passn = 0 Then
    nilaiinisial = nilaiinisial + 1
    Unload Me
    menuisipass.Show
Elseif ID = "0700A184E2C0" And passn <> 0 Then
    nilaiinisial = nilaiinisial + 1
    error = 0
    Unload Me
    menupass.Show
Elseif ID = "0700A10717B6" And passy = 0 Then
    nilaiinisial = nilaiinisial + 2
    Unload Me
    menuisipass.Show
Elseif ID = "0700A10717B6" And passy <> 0 Then
    nilaiinisial = nilaiinisial + 2
    error = 0
    Unload Me
    menupass.Show
Elseif ID = "0700A11915AA" And passa = 0 Then
    nilaiinisial = nilaiinisial + 3
    Unload Me
    menuisipass.Show
Elseif ID = "0700A11915AA" And passa <> 0 Then
    nilaiinisial = nilaiinisial + 3
    error = 0
    Unload Me
    menupass.Show
End If
End Sub

```

Penjelasan dari beberapa baris *coding* tersebut adalah *software* akan memeriksa apakah ID tersebut sudah memiliki *password* atau belum ketika RFID *tag* dibaca pertama kali oleh RFID *reader*. Jika sudah memiliki *password* maka pengguna akan ditampilkan *form* menu verifikasi *password* (menupass), Sedangkan jika belum maka pengguna akan ditampilkan *form*

menu untuk mengisi *password* baru (menuisipass). Berikut merupakan *coding* pada *form* menu *password*.

```

Private Sub bok_Click()
passc = Val(textpass)
If nilaiinisial = 1 And passc = passn Then
    menu_NH.Show
    Unload Me
Elseif nilaiinisial = 2 And passc = passy Then
    menu_SY.Show
    Unload Me
Elseif nilaiinisial = 3 And passc = passa Then
    menu_Admin.Show
    Unload Me
Else
    passsalah.Visible = True
    error = error + 1
    textpass = ""
End If
If error = 3 Then
    error = 0
    Unload Me
    warning.Show
    Select Case nilaiinisial
        Case 1
            user1 = user1 + 1
        Case 2
            user2 = user2 + 1
        Case 3
            user3 = user3 + 1
    End Select
End If
End Sub

```

Pada *coding form* menu *password* pengguna akan diminta memasukan *password* dimana nilai *password* akan dijadikan nilai *passc* sebagai pembanding dengan nilai *password* yang sebelumnya sudah diatur, misalkan *passn*, *passy*, atau *passa*. Kemudian jika nilai *passc* sama dengan nilai *password* dan ID yang bersesuaian maka akan muncul menu pengguna atau administrator. Jika mengalami kesalahan selama tiga kali maka akan muncul menu peringatan (*warning*) dan memberi tahu administrator bahwa terjadi

kesalahan pada saat *log in* dengan memberi nilai 1 pada variabel user1 atau user2 atau user3.

4.2.5. Analisis Hasil Pengujian Penyimpanan dan Pengambilan *Software*

Visual Basic

Prototype ini memiliki 6 buah slot dimana semua slot bisa diakses oleh semua pengguna. Pada Pengujian penyimpanan dan pengambilan *software visual basic* berjalan dengan baik sesuai dengan yang diharapkan. Yang membuat *software* berjalan dengan lancar adalah pengaturan lebih lanjut pada *coding software visual basic*. Yaitu agar slot yang sudah dipilih oleh salah satu pengguna tidak dapat diakses oleh pengguna lain. Ada pun *coding* untuk mengatur hal tersebut adalah sebagai berikut :

1. Pada module

```
Public ts1, ts2, ts3, ts4, ts5, ts6 As Integer
```

Maksud dari ts-n disini adalah id tiap slot yang ada di *prototype*.

2. Pada tombol simpan pengguna 1

```
Public Sub button1ns_Click()
ts1 = ts1 + 1
tanggal1 = Date
jam1 = Time
dataslota = Now
endsimpan.Show
Unload Me
xx = xx + 21.3
data = "@00WJ04480004"
Call ganteng
MSComm1.PortOpen = False
End Sub
```

Pada saat pengguna 1 mengakses slot 1 maka nilai ID ts1 menjadi 1 sehingga tidak dapat diakses oleh pengguna lain yang bernilai ID berbeda.

3. Pada tombol simpan pengguna 2

```
Private Sub button1ss_Click()
ts1 = ts1 + 3
tanggal1 = Date
jam1 = Time
dataslota = Now
endsimpan.Show
Unload Me
xx = xx + 21.3
data = "@00WJ04480004"
Call ganteng
MSComm1.PortOpen = False
End Sub
```

Sedangkan pada saat pengguna 2 mengakses slot 1 maka nilai ID ts1 menjadi 3 sehingga tidak dapat diakses oleh pengguna lain juga.

4. Pada *form* menu ambil pengguna 1

```
If ts1 = 1 Then
  button1na.Visible = True
  tanggalslot1n.Visible = True
  tanggalslot1n.Caption = tanggal1
  janslot1n.Visible = True
  janslot1n.Caption = jam1
Else
  button1na.Visible = False
  tanggalslot1n.Visible = False
  janslot1n.Visible = False
End If
```

Pada menu ambil pengguna 1 software hanya akan memunculkan slot dengan ID bernilai 1. Sehingga jika slot tersebut sudah dipilih oleh pengguna lain maka tidak akan ditampilkan.

5. Pada *form* menu ambil pengguna 2

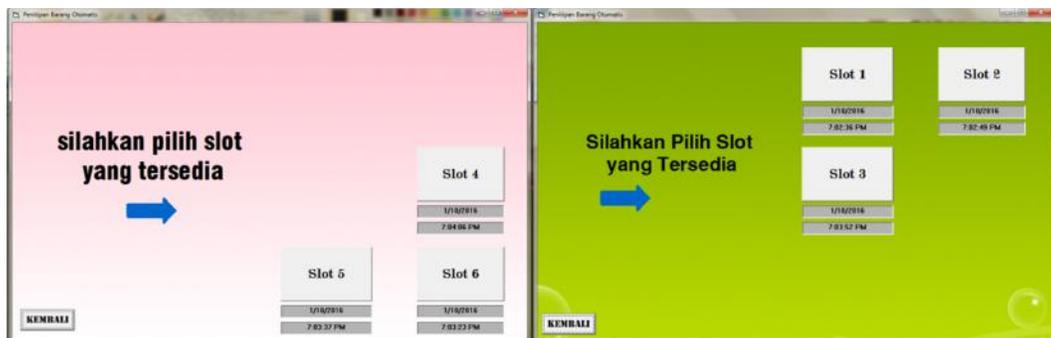
```

If ts1 = 3 Then
  button1sa.Visible = True
  tanggalslot1s.Visible = True
  tanggalslot1s.Caption = tanggal1
  janslot1s.Visible = True
  janslot1s.Caption = jam1
Else
  button1sa.Visible = False
  tanggalslot1s.Visible = False
  janslot1s.Visible = False
End If

```

Pada menu ambil pengguna 2 pun sama, software hanya akan memunculkan slot dengan ID bernilai 3. Sehingga jika slot bernilai 1 seperti pengguna 1 maka slot tersebut tidak akan ditampilkan pada pengguna 2.

Hasil tampilan *coding* pada poin diatas yang diterapkan pada menu ambil ditunjukkan oleh gambar 4.2 berikut.



Gambar 4. 2. Menu Ambil Pengguna

Sumber : Dokumen Pribadi

4.2.6. Analisis Hasil Pengujian Waktu Kerja

Prototype alat ini menggunakan motor 24VDC *gearbox* dengan rasio 1 : 8, setelah itu *output* dari *gearbox* dihubungkan kembali dengan hubung roda. *Prototype* ini memiliki waktu kerja yang lama, dimana semua waktu kerja diatas 30 detik. Waktu kerja tersebut terhitung dari pengguna memilih slot hingga pemindah benda kembali lagi ke posisi awal dan *prototype* siap digunakan kembali.

4.3. Kelebihan dan Kekurangan Alat

4.3.1. Kelebihan Alat

Dari penjelasan dan hasil penelitian yang telah dilakukan, *prototype* ini memiliki beberapa kelebihan, yaitu :

1. Keamanan penyimpanan alat ini sudah terjamin karena memiliki pengaman RFID. Jika seseorang tidak memiliki RFID *tag* yang sesuai dengan daftar RFID yang diperbolehkan maka tidak dapat mengakses alat ini.
2. Disamping menggunakan RFID alat ini juga menggunakan pengaman *password* untuk pengaman lanjutan. Walaupun seseorang memiliki RFID *tag* yang sesuai, ia tidak dapat mengambil benda yang bukan ia simpan karena memiliki keamanan *password* juga.

3. Untuk penyimpanan benda pada slot mana sepenuhnya diatur oleh pengguna itu sendiri dan tidak dapat diakses oleh pengguna lain, sehingga keamanan dan kerahasiaan benda terjamin.
4. Pada saat pengambilan benda terdapat data mengenai benda tersebut disimpan. Metode pengambilan benda dilakukan dengan cara yang sama pada saat penyimpanan.
5. Pada saat darurat terdapat pintu *emergency* yang hanya *administrator* saja bisa membukanya.
6. Ketika terjadi padam listrik *prototype* kembali ke posisi *standby* dengan sendirinya ketika listrik kembali terhubung.
7. Ketika terjadi padam listrik pada *prototype* data penyimpanan tetap tersimpan di dalam *software*. Tapi lain halnya jika PC yang kehilangan daya atau *software* ditutup di PC.

4.3.2. Kekurangan Alat

1. Gerak pemindah benda pada *prototype* ini tergolong lambat. Sehingga memerlukan waktu yang lama untuk menyimpan atau mengambil benda.
2. Karena memerlukan waktu yang cukup lama, dapat terjadi antrian pada saat ingin menggunakan alat ini.
3. Pergerakan masih menimbulkan suara yang tidak nyaman sehingga dapat mengganggu kenyamanan pengguna.
4. Design *prototype* masih kurang sempurna karena masih menggunakan tempat yang besar.

BAB V

KESIMPULAN DAN SARAN

4.1. Kesimpulan

Setelah dilakukan penelitian dengan membuat dan menganalisis *prototype* penyimpanan benda bertingkat otomatis berbasis PLC dengan pengamanan *password* dan RFID menggunakan *interface software visual basic 6.0* maka dapat disimpulkan bahwa :

1. *Prototype* penyimpanan benda bertingkat dapat bekerja secara otomatis dengan menggunakan PLC CJ1M.
2. *Input* yang digunakan sebagai sinyal masukan ke PLC ada 13 *input* yang terdiri atas 12 *limit switch* dan 1 sensor infra merah. Dimana *limit switch* digunakan untuk membatasi pergerakan motor DC, dan sensor infra merah digunakan untuk mendeteksi benda pada pemindah benda.
3. *Output* yang dikendalikan oleh PLC adalah 9 buah relai. Relai pada alat ini digunakan untuk menggerakkan dan merubah putaran motor DC, serta untuk memutus dan menghubungkan daya ke *doorlock* pintu *emergency*.
4. *Software Visual Basic* dapat mengkoneksikan RFID dan PLC sesuai dengan yang diharapkan.
5. Sistem pengaman RFID dan *password* dapat melindungi benda yang disimpan pengguna. Sehingga memberikan rasa aman dan sejalan dengan perkembangan teknologi saat ini.

6. Program pada PLC dengan *interface visual basic* dapat menjalankan sistem penyimpanan benda otomatis sesuai yang diinginkan.

4.2. Saran

Dalam pembuatan *prototype* penyimpanan benda bertingkat otomatis berbasis PLC dengan pengamanan *password* dan RFID ini, penulis menyadari masih banyak kekurangan dan kelemahan pada *prototype* tersebut. Diantaranya adalah sebagai berikut:

1. Pada saat mengkoneksikan RFID dan PLC ke *visual basic* pastikan *commport* dan *comm settings* sudah sesuai dengan cara melakukan pengecekan pada *manage my computer* di komputer atau laptop.
2. Bagian dalam *prototype* masih terkesan kurang rapih karena banyaknya kabel. Oleh karena itu perlu digunakan kabel ukuran kecil tapi dengan kemampuan hantaran dan isolator yang bagus.
3. Suara motor ketika bekerja terdengar keras. Oleh karena itu perlu dibuat sistem mekanis yang lebih halus.
4. Diperlukan sensor tambahan pada tiap slot untuk memastikan slot yang dituju benar-benar kosong ketika akan digunakan untuk menyimpan dan untuk memastikan benda tersimpan atau terambil dengan benar.

DAFTAR PUSTAKA

- Anonim. 2014. *Solenoid Doorlock*. <http://www.geraicerdas.com/motor/solenoid-door-lock-detail>, (diakses pada hari Sabtu, 3 Oktober 2015)
- Crowder, R. M. 1998. *Robot Geometry*.
<http://www.southampton.ac.uk/~rmc1/robotics/argeometry.htm> (diakses pada hari Selasa, 24 Februari 2015)
- Enterprise, Jubile. 2015. *Pemrograman Visual Basic 6, Cara cepat bagi Pemula Menguasai VB 6*. Jakarta : PT. Elex Media Komputindo.
- Parr, E. A. 2003. *Programmable Controllers An Engineer's Guide*. Great Britain: Biddles Ltd.
- Petruzella, Frank D. 2001. *Elektronika Industri. Terj.* Yogyakarta: Andi.
- Pitowarno, Endra. 2006. *Robotika:Desain, Kontrol, dan Kecerdasan Buatan*. Yogyakarta: Andi.
- Praselia, Retna. 2004. *Teori dan Praktek Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0*. Yogyakarta : Andi.
- Putra. *Tool yang Ada didalam VB 6.0*.
<https://hinggakujenuh.wordpress.com/belajar-delphi-7/visualbasic/tool-yang-ada-didalam-vb-6-0/>, (diakses pada hari Jumat, 3 Juli 2015)
- Purnomo, Eko. 2014. *Prinsip Kerja RFID*.
<https://abisabrina.wordpress.com/2014/01/18/prinsip-kerja-RFID/#more-2045>, (diakses pada hari Minggu, 5 Juli 2015)
- Setiawan, Iwan. 2006. *Programmable Logic Controller (PLC) dan Teknik Perancangan Sistem Kontrol*. Yogyakarta: Andi.
- Sulistiono, Ari. 2009. *Antarmuka IED dengan PC melalui MsComm VB6.0*.
<http://ari-sty-blog.blogspot.co.id/2009/11/antarmuka-ied-dengan-pc-melalui-mscomm.html>, (diakses pada hari Sabtu, 4 Juli 2015)
- Syufrijal. 2012. *PLC Konsep, Aplikasi dan Komunikasi Jaringan PLC*. Jakarta.
- Warnock, Ian G. 1988. *PROGRAMMABLE CONTROLLERS, Operation and Application*. Cambridge: Prentice Hall International (UK) Ltd.
- Webb, John W. 1992. *Programmable Logic Controllers*. New York: Macmillan Publishing Company.

Wikipedia. 2014. ***Solenoid***. <https://id.wikipedia.org/wiki/Solenoid>, (diakses pada hari Kamis, 3 Oktober 2015)

ZonaElektro. 2014. ***Sensor Cahaya***. <http://zoniaelektro.net/sensor-cahaya/> (diakses pada hari Selasa, 26 Mei 2015)

LAMPIRAN

1. Coding Program Visual Basic *Prototype*

a. *Form* menu_utama



```
Private Sub Form_Load()
    MSComm1.CommPort = 1
    MSComm1.Settings = "9600,N,8,1"
    MSComm1.InputLen = 0
    MSComm1.RThreshold = 1
    MSComm1.PortOpen = True
End Sub
```

```
Private Sub MSComm1_OnComm()
    If nilaiinisial = 0 Then
        RF = MSComm1.Input
        Text1.Text = Text1.Text & RF
    ElseIf nilaiinisial <> 0 Then
        a = MSComm1.Input
    End If
End Sub
```

```
Private Sub Timer1_Timer()
    sapaan_utama.FontSize = 40
    If Hour(Now) <= 9 Then
        sapaan_utama.Caption = "selamat pagi dan selamat datang di penyimpanan benda otomatis"
    ElseIf Hour(Now) > 9 <= 15 Then
        sapaan_utama.Caption = "selamat siang dan selamat datang di penyimpanan benda otomatis"
    ElseIf Hour(Now) > 15 <= 18 Then
        sapaan_utama.Caption = "selamat petang dan selamat datang di penyimpanan benda otomatis"
    Else
        sapaan_utama.Caption = "selamat malam dan selamat datang di penyimpanan benda otomatis"
    End If
```

```
RS = Text1.Text
ID = Mid(RS, 2, 12)
```

```
If ID = "0700A184E2C0" And passn = 0 Then
    nilaiinisial = nilaiinisial + 1
    Unload Me
    menuisipass.Show
Elseif ID = "0700A184E2C0" And passn <> 0 Then
    nilaiinisial = nilaiinisial + 1
    error = 0
    Unload Me
    menupass.Show
End If
If ID = "0700A184280A" And passr = 0 Then
    nilaiinisial = nilaiinisial + 2
    Unload Me
    menuisipass.Show
Elseif ID = "0700A184280A" And passr <> 0 Then
    nilaiinisial = nilaiinisial + 2
    error = 0
    Unload Me
    menupass.Show
End If
If ID = "0700A10717B6" And passy = 0 Then
    nilaiinisial = nilaiinisial + 3
    Unload Me
    menuisipass.Show
Elseif ID = "0700A10717B6" And passy <> 0 Then
    nilaiinisial = nilaiinisial + 3
    error = 0
    Unload Me
    menupass.Show
End If
If ID = "0700A11915AA" And passa = 0 Then
    nilaiinisial = nilaiinisial + 4
    Unload Me
    menuisipass.Show
Elseif ID = "0700A11915AA" And passa <> 0 Then
    nilaiinisial = nilaiinisial + 4
    error = 0
    Unload Me
    menupass.Show
End If

If (sapaan_utama.Left + sapaan_utama.Width) <= 0 Then
    sapaan_utama.Left = Me.Width
End If
    sapaan_utama.Left = sapaan_utama.Left - 30
End Sub

Private Sub Timer2_Timer()
    If (Image1.Left + Image1.Width) <= 5500 Then
        Image1.Left = 3500
    End If
    Image1.Left = Image1.Left - 30
End Sub
```

b. Form menuisipassbaru



```
Dim pass1 As Single
Dim pass2 As Single
Dim passu As Single
```

```
Private Sub b1_Click()
textpass = textpass + "1"
End Sub
```

```
Private Sub b2_Click()
textpass = textpass + "2"
End Sub
```

```
Private Sub b3_Click()
textpass = textpass + "3"
End Sub
```

```
Private Sub b4_Click()
textpass = textpass + "4"
End Sub
```

```
Private Sub b5_Click()
textpass = textpass + "5"
End Sub
```

```
Private Sub b6_Click()
textpass = textpass + "6"
End Sub
```

```
Private Sub b7_Click()
textpass = textpass + "7"
End Sub
```

```
Private Sub b8_Click()
textpass = textpass + "8"
End Sub
```

```

Private Sub b9_Click()
textpass = textpass + "9"
End Sub

Private Sub b0_Click()
textpass = textpass + "0"
End Sub

Private Sub bcl_Click()
textpass = clc
End Sub

Public Sub bok1_Click()
pass1 = Val(textpass)
textpass = ""
inputulang.Visible = True
Labelpass.Visible = False
passsalah.Visible = False
bok1.Visible = False
bok2.Visible = True
End Sub

Public Sub bok2_Click()
pass2 = Val(textpass)
If pass1 = pass2 Then
    passu = pass2
Else
    passsalah.Visible = True
    bok1.Visible = True
    bok2.Visible = False
    textpass = ""
End If
Select Case nilaiinisial
Case 1
    passn = passn + passu
    Unload Me
    menu_NH.Show
Case 2
    passr = passr + passu
    Unload Me
    menu_RM.Show
Case 3
    passy = passy + passu
    Unload Me
    menu_SY.Show
Case 4
    passa = passa + passu
    Unload Me
    menu_Admin.Show
End Select
End Sub

Private Sub Timerpassi_Timer()
If Len(textpass.Text) < 6 Then
    bok1.Enabled = False

```

```

bok2.Enabled = False
ok.Visible = False
xok.Visible = True
Else
bok1.Enabled = True
bok2.Enabled = True
ok.Visible = True
xok.Visible = False
End If
End Sub

```

c. *Form* menuisipass



Dim passc As Single

```

Private Sub b1_Click()
textpass = textpass + "1"
End Sub

```

```

Private Sub b2_Click()
textpass = textpass + "2"
End Sub

```

```

Private Sub b3_Click()
textpass = textpass + "3"
End Sub

```

```

Private Sub b4_Click()
textpass = textpass + "4"
End Sub

```

```

Private Sub b5_Click()
textpass = textpass + "5"
End Sub

```

```

Private Sub b6_Click()
textpass = textpass + "6"
End Sub

```

```

Private Sub b7_Click()

```

```

textpass = textpass + "7"
End Sub

Private Sub b8_Click()
textpass = textpass + "8"
End Sub

Private Sub b9_Click()
textpass = textpass + "9"
End Sub

Private Sub b0_Click()
textpass = textpass + "0"
End Sub

Private Sub bcl_Click()
textpass = clc
End Sub

Private Sub bok_Click()
passc = Val(textpass)
If nilaiinisial = 1 And passc = passn Then
    menu_NH.Show
    Unload Me
Elseif nilaiinisial = 2 And passc = passr Then
    menu_RM.Show
    Unload Me
Elseif nilaiinisial = 3 And passc = passy Then
    menu_SY.Show
    Unload Me
Elseif nilaiinisial = 4 And passc = passa Then
    menu_Admin.Show
    Unload Me
Else
    passsalah.Visible = True
    error = error + 1
    textpass = ""
End If

If error = 3 Then
    error = 0
    Unload Me
    attention.Show
    Select Case nilaiinisial
        Case 1
            user1 = user1 + 1
        Case 2
            user2 = user2 + 1
        Case 3
            user3 = user3 + 1
    End Select
End If
End Sub

Private Sub Timerpass_Timer()

```

```

If error = 0 Then
    passsalah.Visible = False
End If
If Len(textpass.Text) < 6 Then
    ok.Visible = False
    xok.Visible = True
    bok.Enabled = False
Else
    ok.Visible = True
    xok.Visible = False
    bok.Enabled = True
End If
End Sub

```

d. *Form menu_pengguna1*



```

Public Sub ganteng()
    On Error Resume Next
    Dat$ = data
    s = Len(Dat$)
    r = 0
    For l = 1 To s
        riz$ = Mid$(Dat$, l, 1)
        r = Asc(riz$) Xor r
    Next l
    FCS$ = Hex$(r)
    If Len(FCS$) = 1 Then
        FCS$ = "0" + FCS$
    End If
    DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
    MSComm1.Output = DatTX$
End Sub

```

```

Private Sub exit_Click()
    Unload Me
    menu_utama.Show
    nilaiinisial = 0
End Sub

```

```

Private Sub Form_Load()
    tmn = 0
    MSComm1.CommPort = 2
    MSComm1.Settings = "9600,e,7,2"
End Sub

Private Sub GPS_Click()
    passn = 0
    Unload Me
    menuisipass.Show
End Sub

Public Sub tmlambil_N_Click()
    data = "@00WJ04480002"
    Call ganteng
    Unload Me
    menuambil_NH.Show
End Sub

Public Sub tmlsimpan_N_Click()
    data = "@00WJ04480001"
    Call ganteng
    Unload Me
    menusimpan_NH.Show
    tmns = 0
End Sub

Private Sub Timermnh_Timer()
    If (Image2(1).Top + Image2(1).Height) > 5000 Then
        Image2(1).Top = 3300
    End If
    Image2(1).Top = Image2(1).Top + 50
    If (Image2(0).Top + Image2(0).Height) > 5000 Then
        Image2(0).Top = 3300
    End If
    Image2(0).Top = Image2(0).Top + 50

    tmn = tmn + 1
    If tmn = 1 Then
        MSComm1.PortOpen = True
    End If
    If tmn >= 1 And (ts1 = 1 Or ts2 = 1 Or ts3 = 1 Or ts4 = 1 Or ts5 = 1 Or ts6 = 1)
    Then
        tmlambil_N.Enabled = True
    Else
        tmlambil_N.Enabled = False
    End If
    If tmn >= 1 And (ts1 = 0 Or ts2 = 0 Or ts3 = 0 Or ts4 = 0 Or ts5 = 0 Or ts6 = 0)
    Then
        tmlsimpan_N.Enabled = True
    Else
        tmlsimpan_N.Enabled = False
    End If
End Sub

```

e. *Form menu_pengguna2*



```
Public Sub ganteng()
On Error Resume Next
Dat$ = data
s = Len(Dat$)
r = 0
For I = 1 To s
riz$ = Mid$(Dat$, I, 1)
r = Asc(riz$) Xor r
Next I
FCS$ = Hex$(r)
If Len(FCS$) = 1 Then
FCS$ = "0" + FCS$
End If
DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
MSComm1.Output = DatTX$
End Sub
```

```
Private Sub exit_Click()
Unload Me
menu_utama.Show
nilaiinisial = 0
End Sub
```

```
Private Sub Form_Load()
tms = 0
MSComm1.CommPort = 2
MSComm1.Settings = "9600,e,7,2"
End Sub
```

```
Private Sub GPS_Click()
passy = 0
Unload Me
menuisipass.Show
End Sub
```

```
Public Sub tmlambil_S_Click()
data = "@00WJ04480002"
```

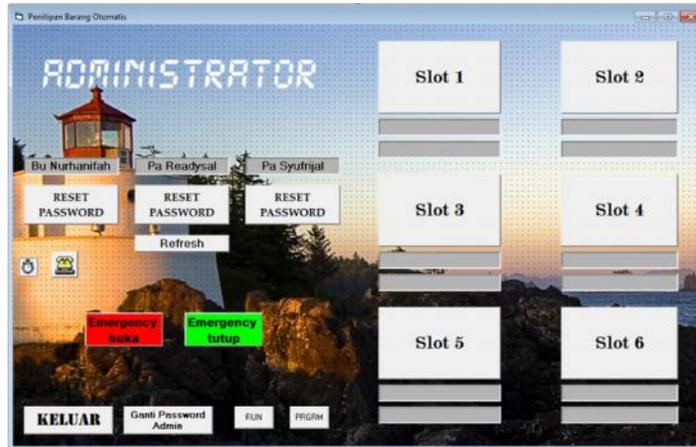
```
Call ganteng
Unload Me
menuambil_SY.Show
End Sub
```

```
Public Sub tmblsimpan_S_Click()
data = "@00WJ04480001"
Call ganteng
Unload Me
menusimpan_SY.Show
tmss = 0
End Sub
```

```
Private Sub Timermsy_Timer()
If (Image2(1).Top + Image2(1).Height) > 5000 Then
    Image2(1).Top = 3300
    End If
    Image2(1).Top = Image2(1).Top + 50
If (Image2(0).Top + Image2(0).Height) > 5000 Then
    Image2(0).Top = 3300
    End If
    Image2(0).Top = Image2(0).Top + 50
```

```
    tms = tms + 1
If tms = 1 Then
    MSComm1.PortOpen = True
    End If
If tms >= 1 And (ts1 = 3 Or ts2 = 3 Or ts3 = 3 Or ts4 = 3 Or ts5 = 3 Or ts6 = 3)
Then
    tmlambil_s.Enabled = True
    Else
    tmlambil_s.Enabled = False
    End If
If tms >= 1 And (ts1 = 0 Or ts2 = 0 Or ts3 = 0 Or ts4 = 0 Or ts5 = 0 Or ts6 = 0)
Then
    tmblsimpan_s.Enabled = True
    Else
    tmblsimpan_s.Enabled = False
    End If
End Sub
```

f. *Form menu_administrator*



```

Public Sub ganteng()
On Error Resume Next
Dat$ = data
s = Len(Dat$)
r = 0
For I = 1 To s
riz$ = Mid$(Dat$, I, 1)
r = Asc(riz$) Xor r
Next I
FCS$ = Hex$(r)
If Len(FCS$) = 1 Then
FCS$ = "0" + FCS$
End If
DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
MSComm1.Output = DatTX$
End Sub

Private Sub button1ma_Click()
ts1 = 0
End Sub

Private Sub button2ma_Click()
ts2 = 0
End Sub

Private Sub button3ma_Click()
ts3 = 0
End Sub

Private Sub button4ma_Click()
ts4 = 0
End Sub

Private Sub button5ma_Click()
ts5 = 0
End Sub

```

```
Private Sub button6ma_Click()  
ts6 = 0  
End Sub
```

```
Private Sub Command1_Click()  
Unload Me  
menuisipass.Show  
tma = 0  
End Sub
```

```
Private Sub emergencybuka_Click()  
data = "@00WJ04490001"  
Call ganteng  
button1ma.Enabled = True  
button2ma.Enabled = True  
button3ma.Enabled = True  
button4ma.Enabled = True  
button5ma.Enabled = True  
button6ma.Enabled = True  
End Sub
```

```
Private Sub emergencytutup_Click()  
data = "@00WJ04490002"  
Call ganteng  
button1ma.Enabled = False  
button2ma.Enabled = False  
button3ma.Enabled = False  
button4ma.Enabled = False  
button5ma.Enabled = False  
button6ma.Enabled = False  
End Sub
```

```
Private Sub exit_Click()  
data = "@00WJ04480000"  
Call ganteng  
data = "@00WJ04490000"  
Call ganteng  
Unload Me  
menu_utama.Show  
nilaiinisial = 0  
End Sub
```

```
Private Sub Form_Load()  
tma = 0  
MSComm1.CommPort = 2  
MSComm1.Settings = "9600,e,7,2"  
End Sub
```

```
Private Sub refresh_Click()  
user1 = user1 - 1  
user2 = user2 - 1  
user3 = user3 - 1  
End Sub
```

```
Private Sub rstuser1_Click()
    passn = 0
    user1 = 0
End Sub

Private Sub rstuser2_Click()
    passr = 0
    user2 = 0
End Sub

Private Sub rstuser3_Click()
    passy = 0
    user3 = 0
End Sub

Private Sub run_Click()
    data = "@00SC02"
    Call ganteng
End Sub

Private Sub program_Click()
    data = "@00SC00"
    Call ganteng
End Sub

Private Sub Timeradm_Timer()
    Select Case ts1
    Case 0
        userslot1.Caption = "Kosong"
        userslot1.BackColor = &HFF00&
        dataslot1.Caption = ""
        dataslot1.BackColor = &HFF00&
    Case 1
        userslot1.Caption = "Bu Nurhanifah"
        userslot1.BackColor = &HFF&
        dataslot1.Caption = dataslota
        dataslot1.BackColor = &HFF&
    Case 2
        userslot1.Caption = "Pengguna 3"
        userslot1.BackColor = &HFF&
        dataslot1.Caption = dataslota
        dataslot1.BackColor = &HFF&
    Case 3
        userslot1.Caption = "Pa Syufrijal"
        userslot1.BackColor = &HFF&
        dataslot1.Caption = dataslota
        dataslot1.BackColor = &HFF&
    End Select

    Select Case ts2
    Case 0
        userslot2.Caption = "Kosong"
        userslot2.BackColor = &HFF00&
        dataslot2.Caption = ""
        dataslot2.BackColor = &HFF00&
```

```
Case 1
  userslot2.Caption = "Bu Nurhanifah"
  userslot2.BackColor = &HFF&
  dataslot2.Caption = dataslotb
  dataslot2.BackColor = &HFF&
Case 2
  userslot2.Caption = "Pengguna 3"
  userslot2.BackColor = &HFF&
  dataslot2.Caption = dataslotb
  dataslot2.BackColor = &HFF&
Case 3
  userslot2.Caption = "Pa Syufrijal"
  userslot2.BackColor = &HFF&
  dataslot2.Caption = dataslotb
  dataslot2.BackColor = &HFF&
End Select
```

```
Select Case ts3
```

```
Case 0
  userslot3.Caption = "Kosong"
  userslot3.BackColor = &HFF00&
  dataslot3.Caption = ""
  dataslot3.BackColor = &HFF00&
Case 1
  userslot3.Caption = "Bu Nurhanifah"
  userslot3.BackColor = &HFF&
  dataslot3.Caption = dataslotc
  dataslot3.BackColor = &HFF&
Case 2
  userslot3.Caption = "Pengguna 3"
  userslot3.BackColor = &HFF&
  dataslot3.Caption = dataslotc
  dataslot3.BackColor = &HFF&
Case 3
  userslot3.Caption = "Pa Syufrijal"
  userslot3.BackColor = &HFF&
  dataslot3.Caption = dataslotc
  dataslot3.BackColor = &HFF&
End Select
```

```
Select Case ts4
```

```
Case 0
  userslot4.Caption = "Kosong"
  userslot4.BackColor = &HFF00&
  dataslot4.Caption = ""
  dataslot4.BackColor = &HFF00&
Case 1
  userslot4.Caption = "Bu Nurhanifah"
  userslot4.BackColor = &HFF&
  dataslot4.Caption = dataslotd
  dataslot4.BackColor = &HFF&
Case 2
  userslot4.Caption = "Pengguna 3"
  userslot4.BackColor = &HFF&
  dataslot4.Caption = dataslotd
```

```
    dataslot4.BackColor = &HFF&
Case 3
    userslot4.Caption = "Pa Syufrijal"
    userslot4.BackColor = &HFF&
    dataslot4.Caption = dataslotd
    dataslot4.BackColor = &HFF&
End Select

Select Case ts5
Case 0
    userslot5.Caption = "Kosong"
    userslot5.BackColor = &HFF00&
    dataslot5.Caption = ""
    dataslot5.BackColor = &HFF00&
Case 1
    userslot5.Caption = "Bu Nurhanifah"
    userslot5.BackColor = &HFF&
    dataslot5.Caption = dataslote
    dataslot5.BackColor = &HFF&
Case 2
    userslot5.Caption = "Pengguna 3"
    userslot5.BackColor = &HFF&
    dataslot5.Caption = dataslote
    dataslot5.BackColor = &HFF&
Case 3
    userslot5.Caption = "Pa Syufrijal"
    userslot5.BackColor = &HFF&
    dataslot5.Caption = dataslote
    dataslot5.BackColor = &HFF&
End Select

Select Case ts6
Case 0
    userslot6.Caption = "Kosong"
    userslot6.BackColor = &HFF00&
    dataslot6.Caption = ""
    dataslot6.BackColor = &HFF00&
Case 1
    userslot6.Caption = "Bu Nurhanifah"
    userslot6.BackColor = &HFF&
    dataslot6.Caption = dataslotf
    dataslot6.BackColor = &HFF&
Case 2
    userslot6.Caption = "Pengguna 3"
    userslot6.BackColor = &HFF&
    dataslot6.Caption = dataslotf
    dataslot6.BackColor = &HFF&
Case 3
    userslot6.Caption = "Pa Syufrijal"
    userslot6.BackColor = &HFF&
    dataslot6.Caption = dataslotf
    dataslot6.BackColor = &HFF&
End Select
```

```

tma = tma + 1
If tma = 1 Then
    MSComm1.PortOpen = True
End If

If user1 = 0 Then
    Labeluser1.BackColor = &H8000000A
Elseif user1 >= 1 Then
    Labeluser1.BackColor = &HFF&
End If

If user2 = 0 Then
    Labeluser2.BackColor = &H8000000A
Elseif user2 >= 1 Then
    Labeluser2.BackColor = &HFF&
End If

If user3 = 0 Then
    Labeluser3.BackColor = &H8000000A
Elseif user3 >= 1 Then
    Labeluser3.BackColor = &HFF&
End If
End Sub

```

g. Form menu_simpanpengguna1



```

Public Sub ganteng()
On Error Resume Next
Dat$ = data
s = Len(Dat$)
r = 0
For I = 1 To s
riz$ = Mid$(Dat$, I, 1)
r = Asc(riz$) Xor r
Next I
FCS$ = Hex$(r)
If Len(FCS$) = 1 Then
FCS$ = "0" + FCS$

```

```
End If
DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
MSComm1.Output = DatTX$
End Sub
```

```
Private Sub Form_Load()
    MSComm1.CommPort = 2
    MSComm1.Settings = "9600,E,7,2"
End Sub
```

```
Public Sub button1ns_Click()
    ts1 = ts1 + 1
    tanggal1 = Date
    jam1 = Time
    dataslota = Now
    Timer1.Enabled = False
    data = "@00WJ04480004"
    Call ganteng
    Unload Me
    endsimpan.Show
End Sub
```

```
Public Sub button2ns_Click()
    ts2 = ts2 + 1
    tanggal2 = Date
    jam2 = Time
    dataslotb = Now
    Timer1.Enabled = False
    data = "@00WJ04480008"
    Call ganteng
    Unload Me
    endsimpan.Show
End Sub
```

```
Public Sub button3ns_Click()
    ts3 = ts3 + 1
    tanggal3 = Date
    jam3 = Time
    dataslotc = Now
    Timer1.Enabled = False
    data = "@00WJ04480010"
    Call ganteng
    Unload Me
    endsimpan.Show
End Sub
```

```
Public Sub button4ns_Click()
    ts4 = ts4 + 1
    tanggal4 = Date
    jam4 = Time
    dataslotd = Now
    Timer1.Enabled = False
    data = "@00WJ04480020"
    Call ganteng
    Unload Me
```

```
endsimpan.Show
End Sub

Public Sub button5ns_Click()
ts5 = ts5 + 1
tanggal5 = Date
jam5 = Time
dataslote = Now
Timer1.Enabled = False
data = "@00WJ04480040"
Call ganteng
Unload Me
endsimpan.Show
End Sub

Public Sub button6ns_Click()
ts6 = ts6 + 1
tanggal6 = Date
jam6 = Time
dataslotf = Now
Timer1.Enabled = False
data = "@00WJ04480080"
Call ganteng
Unload Me
endsimpan.Show
End Sub

Private Sub kembali_Click()
data = "@00WJ04480000"
Call ganteng
Timer1.Enabled = False
MSComm1.PortOpen = False
Unload Me
menu_NH.Show
End Sub

Private Sub Timersnh_Timer()
tmns = tmns + 1
If tmns = 1 Then
    MSComm1.PortOpen = True
End If

If tmns = 1 Then
Timer1.Enabled = True
End If

If ts1 = 0 Then
    button1ns.Visible = True
Else
    button1ns.Visible = False
End If
If ts2 = 0 Then
    button2ns.Visible = True
Else
    button2ns.Visible = False
```

```
End If
If ts3 = 0 Then
    button3ns.Visible = True
Else
    button3ns.Visible = False
End If
If ts4 = 0 Then
    button4ns.Visible = True
Else
    button4ns.Visible = False
End If
If ts5 = 0 Then
    button5ns.Visible = True
Else
    button5ns.Visible = False
End If
If ts6 = 0 Then
    button6ns.Visible = True
Else
    button6ns.Visible = False
End If

If (arrowr_NHS.Left + arrowr_NHS.Width) > 6000 Then
    arrowr_NHS.Left = 1500
End If
    arrowr_NHS.Left = arrowr_NHS.Left + 150
End Sub

Private Sub Timer1_Timer()
    data = "@00RR00080001"
    Call ganteng
    iki = MSComm1.Input
    ipah$ = Mid(iki, 8, 4)
    syf = ipah$
    Select Case syf
        Case "0000"
            button1ns.Enabled = False
            button2ns.Enabled = False
            button3ns.Enabled = False
            button4ns.Enabled = False
            button5ns.Enabled = False
            button6ns.Enabled = False
            kembali.Enabled = True
        Case "0001"
            button1ns.Enabled = True
            button2ns.Enabled = True
            button3ns.Enabled = True
            button4ns.Enabled = True
            button5ns.Enabled = True
            button6ns.Enabled = True
            kembali.Enabled = False
    End Select
End Sub
```

h. Form menu_simpanpengguna2



```
Public Sub ganteng()
On Error Resume Next
Dat$ = data
s = Len(Dat$)
r = 0
For I = 1 To s
riz$ = Mid$(Dat$, I, 1)
r = Asc(riz$) Xor r
Next I
FCS$ = Hex$(r)
If Len(FCS$) = 1 Then
FCS$ = "0" + FCS$
End If
DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
MSComm1.Output = DatTX$
End Sub
```

```
Private Sub Form_Load()
MSComm1.CommPort = 2
MSComm1.Settings = "9600,e,7,2"
End Sub
```

```
Private Sub button1ss_Click()
ts1 = ts1 + 3
tanggal1 = Date
jam1 = Time
dataslota = Now
Timer1.Enabled = False
data = "@00WJ04480004"
Call ganteng
Unload Me
endsimpan.Show
End Sub
```

```
Private Sub button2ss_Click()
ts2 = ts2 + 3
tanggal2 = Date
```

```
jam2 = Time
dataslotb = Now
Timer1.Enabled = False
data = "@00WJ04480008"
Call ganteng
Unload Me
endsimpan.Show
End Sub
```

```
Private Sub button3ss_Click()
ts3 = ts3 + 3
tanggal3 = Date
jam3 = Time
dataslotc = Now
Timer1.Enabled = False
data = "@00WJ04480010"
Call ganteng
Unload Me
endsimpan.Show
End Sub
```

```
Private Sub button4ss_Click()
ts4 = ts4 + 3
tanggal4 = Date
jam4 = Time
dataslotd = Now
Timer1.Enabled = False
data = "@00WJ04480020"
Call ganteng
Unload Me
endsimpan.Show
End Sub
```

```
Private Sub button5ss_Click()
ts5 = ts5 + 3
tanggal5 = Date
jam5 = Time
dataslote = Now
Timer1.Enabled = False
data = "@00WJ04480040"
Call ganteng
Unload Me
endsimpan.Show
End Sub
```

```
Private Sub button6ss_Click()
ts6 = ts6 + 3
tanggal6 = Date
jam6 = Time
dataslotf = Now
Timer1.Enabled = False
data = "@00WJ04480080"
Call ganteng
Unload Me
endsimpan.Show
```

```
End Sub
```

```
Private Sub kembali_Click()  
data = "@00WJ04480000"  
Call ganteng  
Timer1.Enabled = False  
MSComm1.PortOpen = False  
Unload Me  
menu_SY.Show  
End Sub
```

```
Private Sub Timerssy_Timer()  
tmss = tmss + 1  
If tmss = 1 Then  
    MSComm1.PortOpen = True  
End If
```

```
If tmss = 1 Then  
Timer1.Enabled = True  
End If
```

```
If ts1 = 0 Then  
    button1ss.Visible = True  
Else  
    button1ss.Visible = False  
End If
```

```
If ts2 = 0 Then  
    button2ss.Visible = True  
Else  
    button2ss.Visible = False  
End If
```

```
If ts3 = 0 Then  
    button3ss.Visible = True  
Else  
    button3ss.Visible = False  
End If
```

```
If ts4 = 0 Then  
    button4ss.Visible = True  
Else  
    button4ss.Visible = False  
End If
```

```
If ts5 = 0 Then  
    button5ss.Visible = True  
Else  
    button5ss.Visible = False  
End If
```

```
If ts6 = 0 Then  
    button6ss.Visible = True  
Else  
    button6ss.Visible = False  
End If
```

```
If (arrowr_SYS.Left + arrowr_SYS.Width) > 15000 Then  
    arrowr_SYS.Left = 1500  
End If
```

```

arrowr_SYS.Left = arrowr_SYS.Left + 150
End Sub

```

```

Private Sub Timer1_Timer()
data = "@00RR00080001"
Call ganteng
iki = MSComm1.Input
ipah$ = Mid(iki, 8, 4)
syf = ipah$
Select Case syf
Case "0000"
button1ss.Enabled = False
button2ss.Enabled = False
button3ss.Enabled = False
button4ss.Enabled = False
button5ss.Enabled = False
button6ss.Enabled = False
kembali.Enabled = True
Case "0001"
button1ss.Enabled = True
button2ss.Enabled = True
button3ss.Enabled = True
button4ss.Enabled = True
button5ss.Enabled = True
button6ss.Enabled = True
kembali.Enabled = False
End Select
End Sub

```

i. Form menu_ambilpengguna1



```

Public Sub ganteng()
On Error Resume Next
Dat$ = data
s = Len(Dat$)
r = 0
For I = 1 To s
riz$ = Mid$(Dat$, I, 1)
r = Asc(riz$) Xor r

```

```
Next I
FCS$ = Hex$(r)
If Len(FCS$) = 1 Then
FCS$ = "0" + FCS$
End If
DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
MSComm1.Output = DatTX$
End Sub
```

```
Private Sub Form_Load()
    MSComm1.CommPort = 2
    MSComm1.Settings = "9600,E,7,2"
    MSComm1.PortOpen = True
End Sub
```

```
Private Sub button1na_Click()
ts1 = ts1 - 1
dataslot1 = ""
data = "@00WJ04480200"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
```

```
End Sub
```

```
Private Sub button2na_Click()
ts2 = ts2 - 1
dataslot2 = ""
data = "@00WJ04480400"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub button3na_Click()
ts3 = ts3 - 1
dataslot3 = ""
data = "@00WJ04480800"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub button4na_Click()
ts4 = ts4 - 1
dataslot4 = ""
data = "@00WJ04481000"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub button5na_Click()
ts5 = ts5 - 1
dataslot5 = ""
data = "@00WJ04482000"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub

Private Sub button6na_Click()
ts6 = ts6 - 1
dataslot6 = ""
data = "@00WJ04484000"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub

Private Sub kembali_Click()
data = "@00WJ04480000"
Call ganteng
tmn = 0
MSComm1.PortOpen = False
Unload Me
menu_NH.Show
End Sub

Private Sub Timeranh_Timer()
If ts1 = 1 Then
    button1na.Visible = True
    tanggalslot1n.Visible = True
    tanggalslot1n.Caption = tanggal1
    janslot1n.Visible = True
    janslot1n.Caption = jam1
Else
    button1na.Visible = False
    tanggalslot1n.Visible = False
    janslot1n.Visible = False
End If
If ts2 = 1 Then
    button2na.Visible = True
    tanggalslot2n.Visible = True
    tanggalslot2n.Caption = tanggal2
    janslot2n.Visible = True
    janslot2n.Caption = jam2
Else
    button2na.Visible = False
    tanggalslot2n.Visible = False
    janslot2n.Visible = False
End If
If ts3 = 1 Then
    button3na.Visible = True
```

```
tanggalslot3n.Visible = True
tanggalslot3n.Caption = tanggal3
jamslot3n.Visible = True
jamslot3n.Caption = jam3
Else
button3na.Visible = False
tanggalslot3n.Visible = False
jamslot3n.Visible = False
End If
If ts4 = 1 Then
button4na.Visible = True
tanggalslot4n.Visible = True
tanggalslot4n.Caption = tanggal4
jamslot4n.Visible = True
jamslot4n.Caption = jam4
Else
button4na.Visible = False
tanggalslot4n.Visible = False
jamslot4n.Visible = False
End If
If ts5 = 1 Then
button5na.Visible = True
tanggalslot5n.Visible = True
tanggalslot5n.Caption = tanggal5
jamslot5n.Visible = True
jamslot5n.Caption = jam5
Else
button5na.Visible = False
tanggalslot5n.Visible = False
jamslot5n.Visible = False
End If
If ts6 = 1 Then
button6na.Visible = True
tanggalslot6n.Visible = True
tanggalslot6n.Caption = tanggal6
jamslot6n.Visible = True
jamslot6n.Caption = jam6
Else
button6na.Visible = False
tanggalslot6n.Visible = False
jamslot6n.Visible = False
End If

arrowr_NHA.Left = arrowr_NHA.Left + 150
If (arrowr_NHA.Left + arrowr_NHA.Width) > 6000 Then
arrowr_NHA.Left = 1500
End If
End Sub
```

j. *Form menu_ambilpengguna2*



```
Public Sub ganteng()
On Error Resume Next
Dat$ = data
s = Len(Dat$)
r = 0
For l = 1 To s
riz$ = Mid$(Dat$, l, 1)
r = Asc(riz$) Xor r
Next l
FCS$ = Hex$(r)
If Len(FCS$) = 1 Then
FCS$ = "0" + FCS$
End If
DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
MSComm1.Output = DatTX$
End Sub
```

```
Private Sub Form_Load()
MSComm1.CommPort = 2
MSComm1.Settings = "9600,e,7,2"
MSComm1.PortOpen = True
End Sub
```

```
Private Sub button1sa_Click()
ts1 = ts1 - 3
dataslot1 = ""
data = "@00WJ04480200"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub button2sa_Click()
ts2 = ts2 - 3
dataslot2 = ""
data = "@00WJ04480400"
```

```
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub button3sa_Click()
ts3 = ts3 - 3
dataslot3 = ""
data = "@00WJ04480800"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub button4sa_Click()
ts4 = ts4 - 3
dataslot4 = ""
data = "@00WJ04481000"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub button5sa_Click()
ts5 = ts5 - 3
dataslot5 = ""
data = "@00WJ04482000"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub button6sa_Click()
ts6 = ts6 - 3
dataslot6 = ""
data = "@00WJ04484000"
Call ganteng
MSComm1.PortOpen = False
Unload Me
endambil.Show
End Sub
```

```
Private Sub kembali_Click()
data = "@00WJ04480000"
Call ganteng
tms = 0
MSComm1.PortOpen = False
Unload Me
menu_SY.Show
End Sub
```

```
Private Sub Timerasy_Timer()  
If ts1 = 3 Then  
    button1sa.Visible = True  
    tanggalslot1s.Visible = True  
    tanggalslot1s.Caption = tanggal1  
    jamslot1s.Visible = True  
    jamslot1s.Caption = jam1  
Else  
    button1sa.Visible = False  
    tanggalslot1s.Visible = False  
    jamslot1s.Visible = False  
End If  
If ts2 = 3 Then  
    button2sa.Visible = True  
    tanggalslot2s.Visible = True  
    tanggalslot2s.Caption = tanggal2  
    jamslot2s.Visible = True  
    jamslot2s.Caption = jam2  
Else  
    button2sa.Visible = False  
    tanggalslot2s.Visible = False  
    jamslot2s.Visible = False  
End If  
If ts3 = 3 Then  
    button3sa.Visible = True  
    tanggalslot3s.Visible = True  
    tanggalslot3s.Caption = tanggal3  
    jamslot3s.Visible = True  
    jamslot3s.Caption = jam3  
Else  
    button3sa.Visible = False  
    tanggalslot3s.Visible = False  
    jamslot3s.Visible = False  
End If  
If ts4 = 3 Then  
    button4sa.Visible = True  
    tanggalslot4s.Visible = True  
    tanggalslot4s.Caption = tanggal4  
    jamslot4s.Visible = True  
    jamslot4s.Caption = jam4  
Else  
    button4sa.Visible = False  
    tanggalslot4s.Visible = False  
    jamslot4s.Visible = False  
End If  
If ts5 = 3 Then  
    button5sa.Visible = True  
    tanggalslot5s.Visible = True  
    tanggalslot5s.Caption = tanggal5  
    jamslot5s.Visible = True  
    jamslot5s.Caption = jam5  
Else  
    button5sa.Visible = False  
    tanggalslot5s.Visible = False  
    jamslot5s.Visible = False  
End If  
End Sub
```

```

End If
If ts6 = 3 Then
    button6sa.Visible = True
    tanggalslot6s.Visible = True
    tanggalslot6s.Caption = tanggal6
    jamslot6s.Visible = True
    jamslot6s.Caption = jam6
Else
    button6sa.Visible = False
    tanggalslot6s.Visible = False
    jamslot6s.Visible = False
End If

If (arrowr_SYA.Left + arrowr_SYA.Width) > 6000 Then
    arrowr_SYA.Left = 1500
End If
    arrowr_SYA.Left = arrowr_SYA.Left + 150

End Sub

```

k. Form akhir_simpan



```

Public Sub ganteng()
    On Error Resume Next
    Dat$ = data
    s = Len(Dat$)
    r = 0
    For I = 1 To s
        riz$ = Mid$(Dat$, I, 1)
        r = Asc(riz$) Xor r
    Next I
    FCS$ = Hex$(r)
    If Len(FCS$) = 1 Then
        FCS$ = "0" + FCS$
    End If
    DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
    MSComm1.Output = DatTX$
End Sub

```

```

Private Sub Form_Load()
tmea = 0
  MSComm1.CommPort = 2
  MSComm1.Settings = "9600,e,7,2"
End Sub

```

```

Private Sub Timerend_Timer()
tmea = tmea + 1
If tmea = 3 Then
MSComm1.PortOpen = True
Timer2.Enabled = True
End If
End Sub

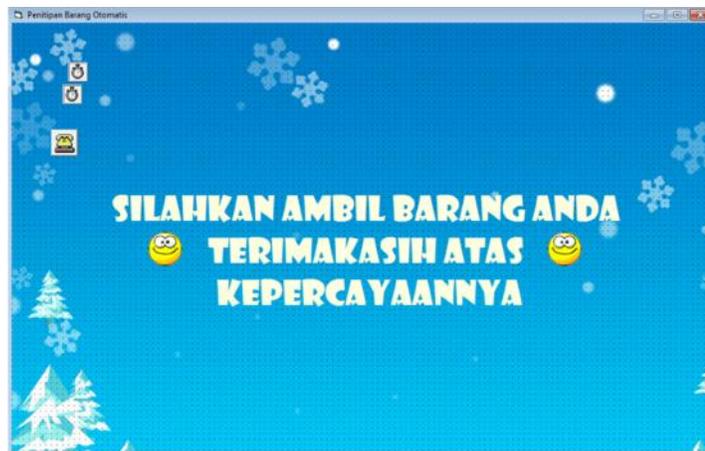
```

```

Private Sub Timer2_Timer()
data = "@00RR00100001"
Call ganteng
iki = MSComm1.Input
ipah$ = Mid(iki, 8, 4)
syf = ipah$
If syf = "0001" Then
  nilaiinisial = 0
  Timer2.Enabled = False
  MSComm1.PortOpen = False
  Unload Me
  menu_utama.Show
End If
End Sub

```

1. *Form akhir_ambil*



```

Public Sub ganteng()
On Error Resume Next
Dat$ = data
s = Len(Dat$)
r = 0
For l = 1 To s

```

```
riz$ = Mid$(Dat$, I, 1)
r = Asc(riz$) Xor r
Next I
FCS$ = Hex$(r)
If Len(FCS$) = 1 Then
FCS$ = "0" + FCS$
End If
DatTX$ = Dat$ + FCS$ + "*" + Chr$(13)
MSComm1.Output = DatTX$
End Sub
```

```
Private Sub Form_Load()
tme = 0
    MSComm1.CommPort = 2
    MSComm1.Settings = "9600,e,7,2"
End Sub
```

```
Private Sub Timer1_Timer()
tme = tme + 1
If tme = 3 Then
MSComm1.PortOpen = True
Timer2.Enabled = True
End If
End Sub
```

```
Private Sub Timer2_Timer()
data = "@00RR00090001"
Call ganteng
iki = MSComm1.Input
ipah$ = Mid(iki, 8, 4)
syf = ipah$
If syf = "0001" Then
    nilaiinisial = 0
    Timer2.Enabled = False
    MSComm1.PortOpen = False
    Unload Me
    menu_utama.Show
End If
End Sub
```

m. Form peringatan



```

Private Sub Timerend_Timer()
Static tend As Integer
If tend <= 100 Then
loding.Value = tend
tend = tend + 5
Else
If loding.Max = 100 Then
tend = 0
nilaiinisial = 0
menu_utama.Show
Unload Me
End If
End If
End Sub

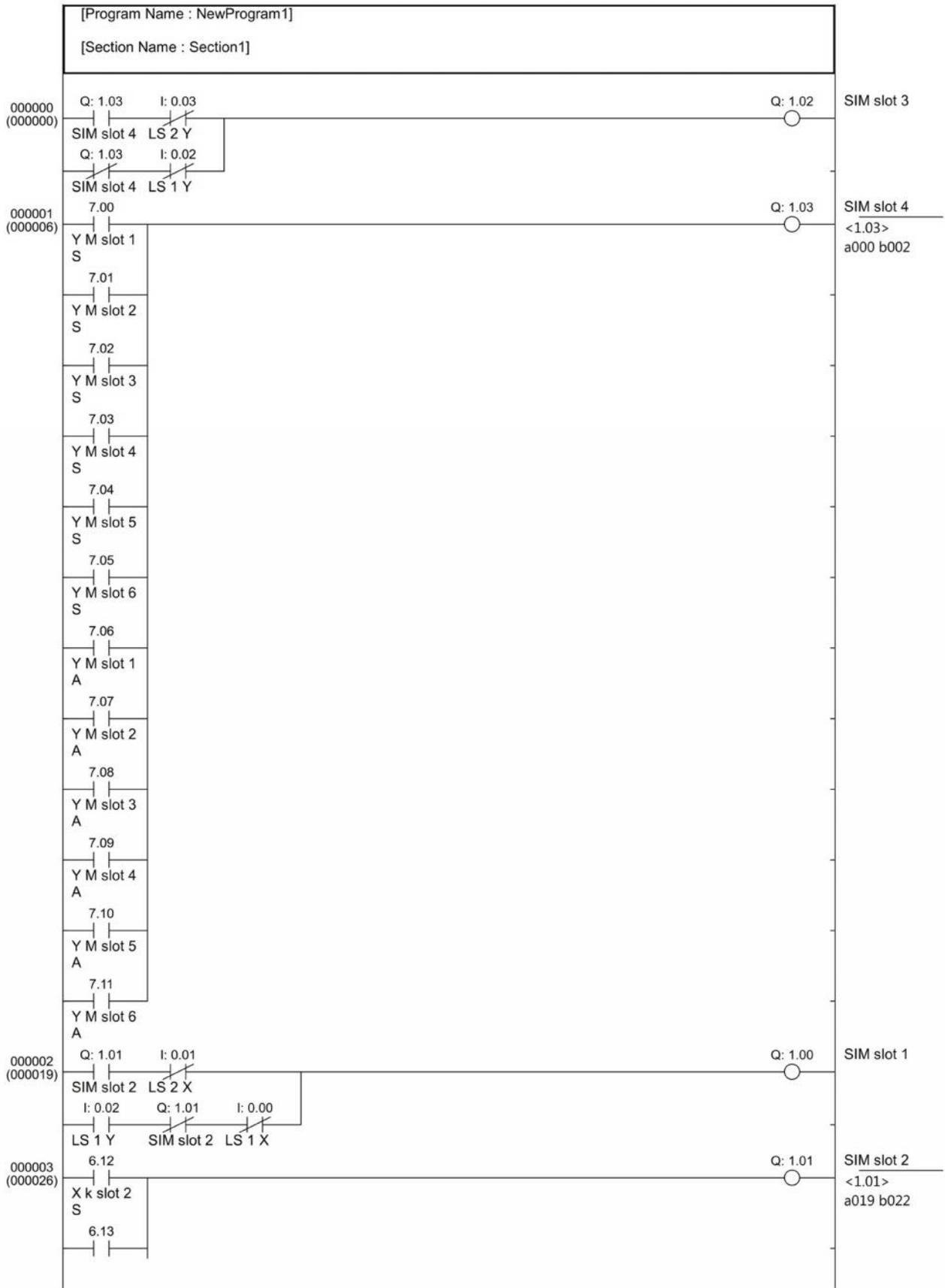
```

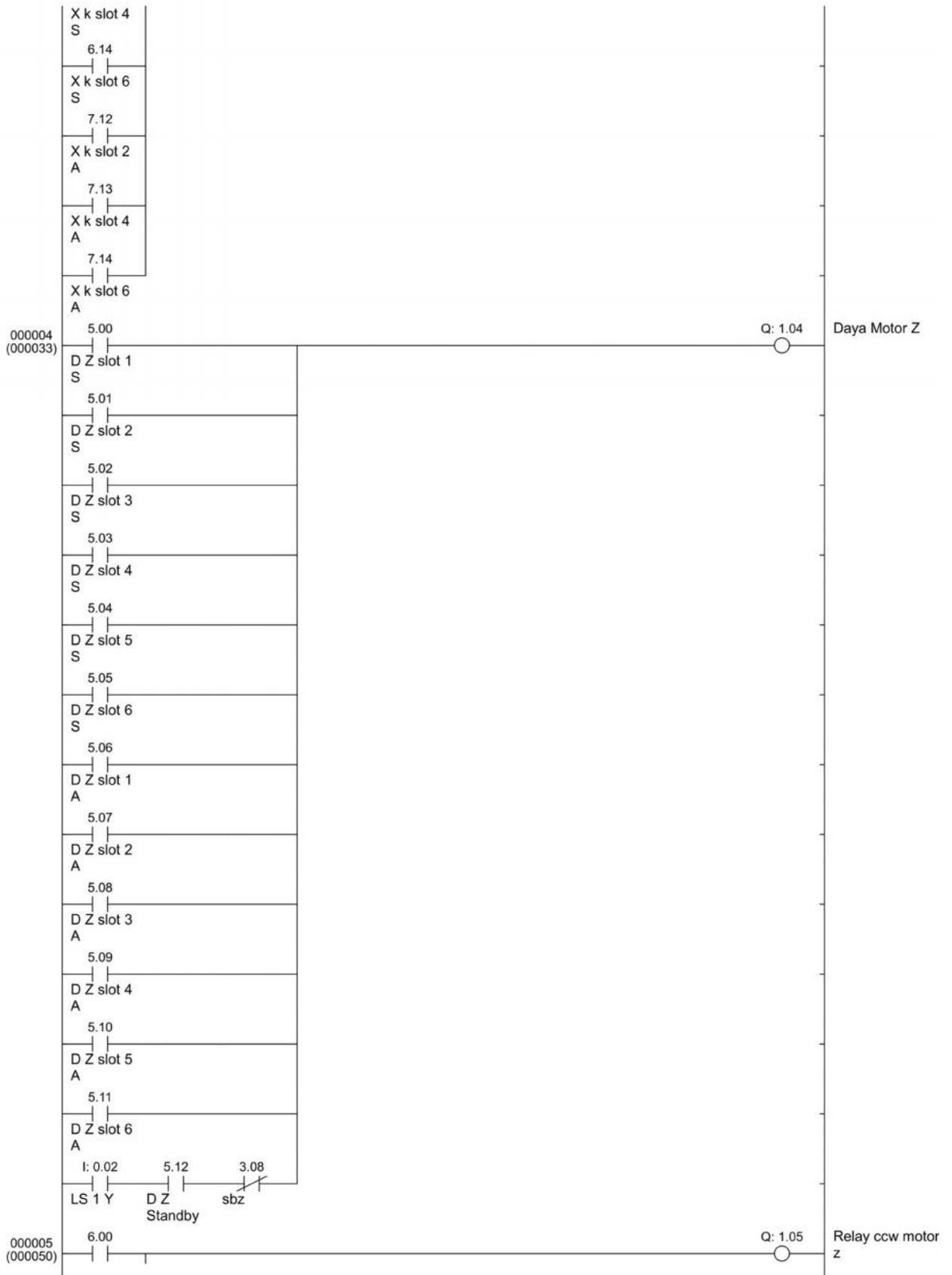
n. Module

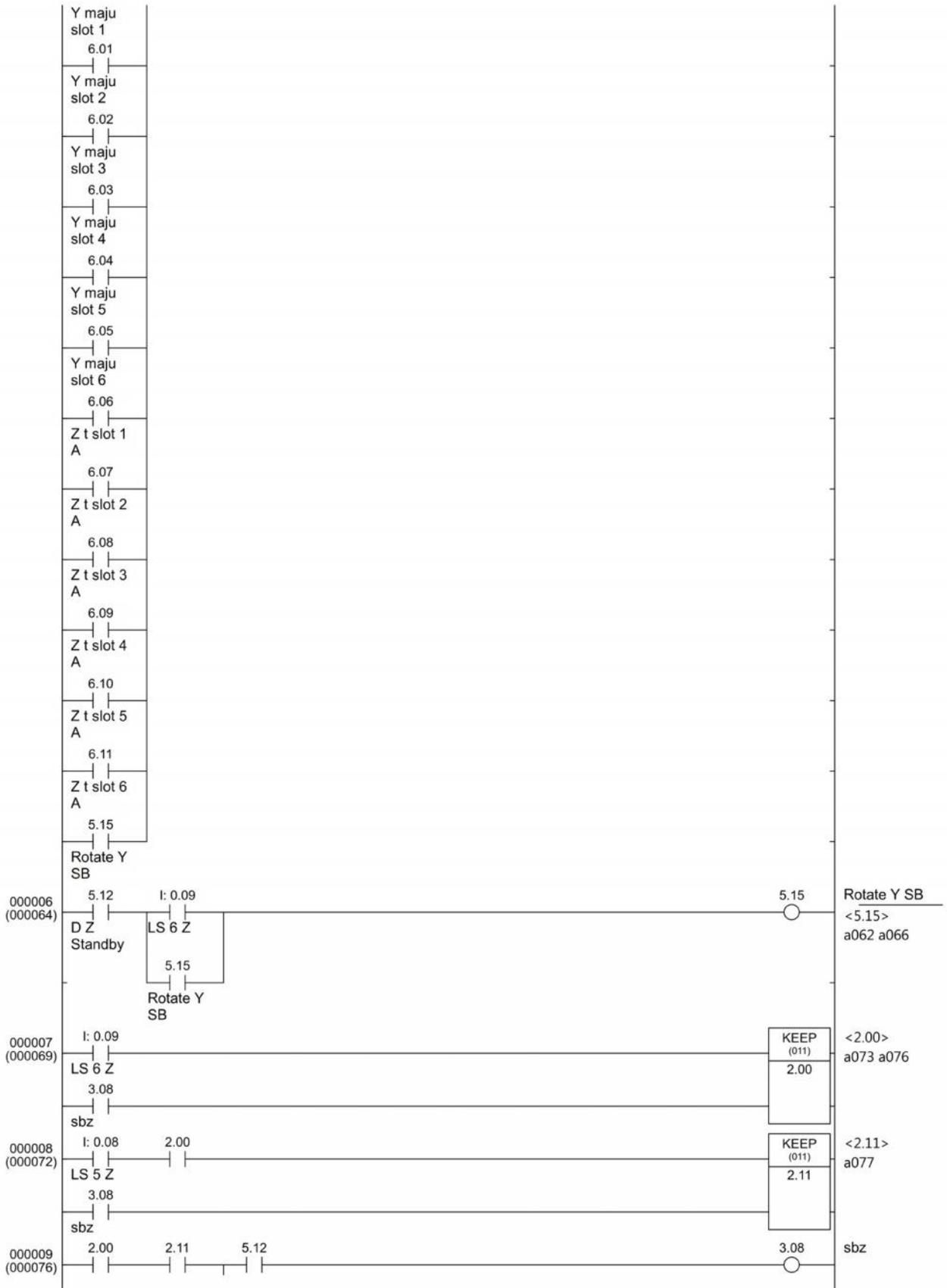
```

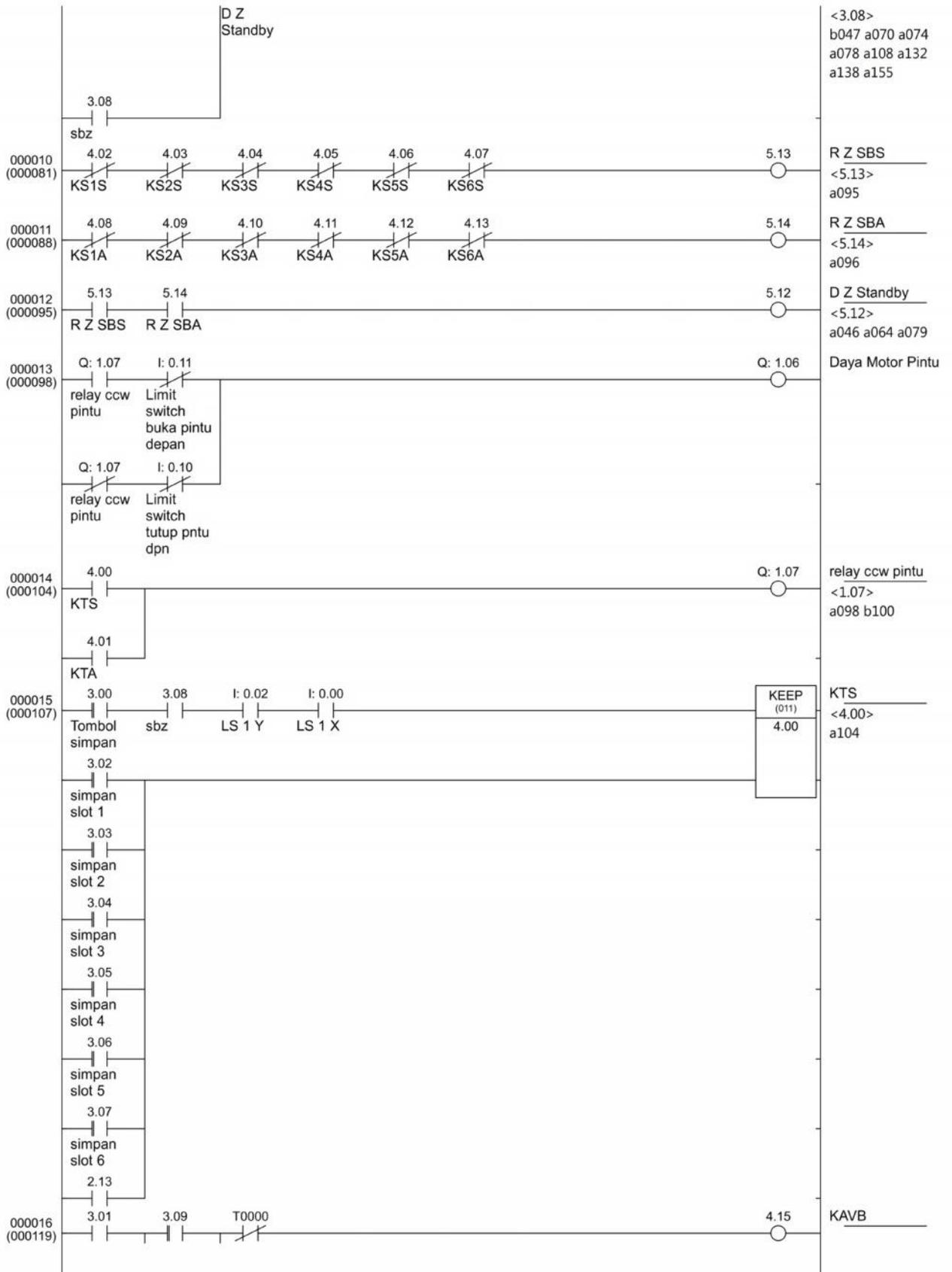
Public ts1, ts2, ts3, ts4, ts5, ts6 As Integer
Public tanggal1, tanggal2, tanggal3, tanggal4, tanggal5, tanggal6 As String
Public jam1, jam2, jam3, jam4, jam5, jam6 As String
Public dataslota, dataslotb, dataslotc, dataslotd, dataslote, dataslotf As String
Public passn, passr, passy, passa As Single
Public error As Integer
Public nilaiinisial As Integer
Public user1, user2, user3 As Single
Public RF, a As String
Public data As String
Public tmu, tms, tmn, tmr, tma, tme, tmea, tmns, tmrs, tmss As Long

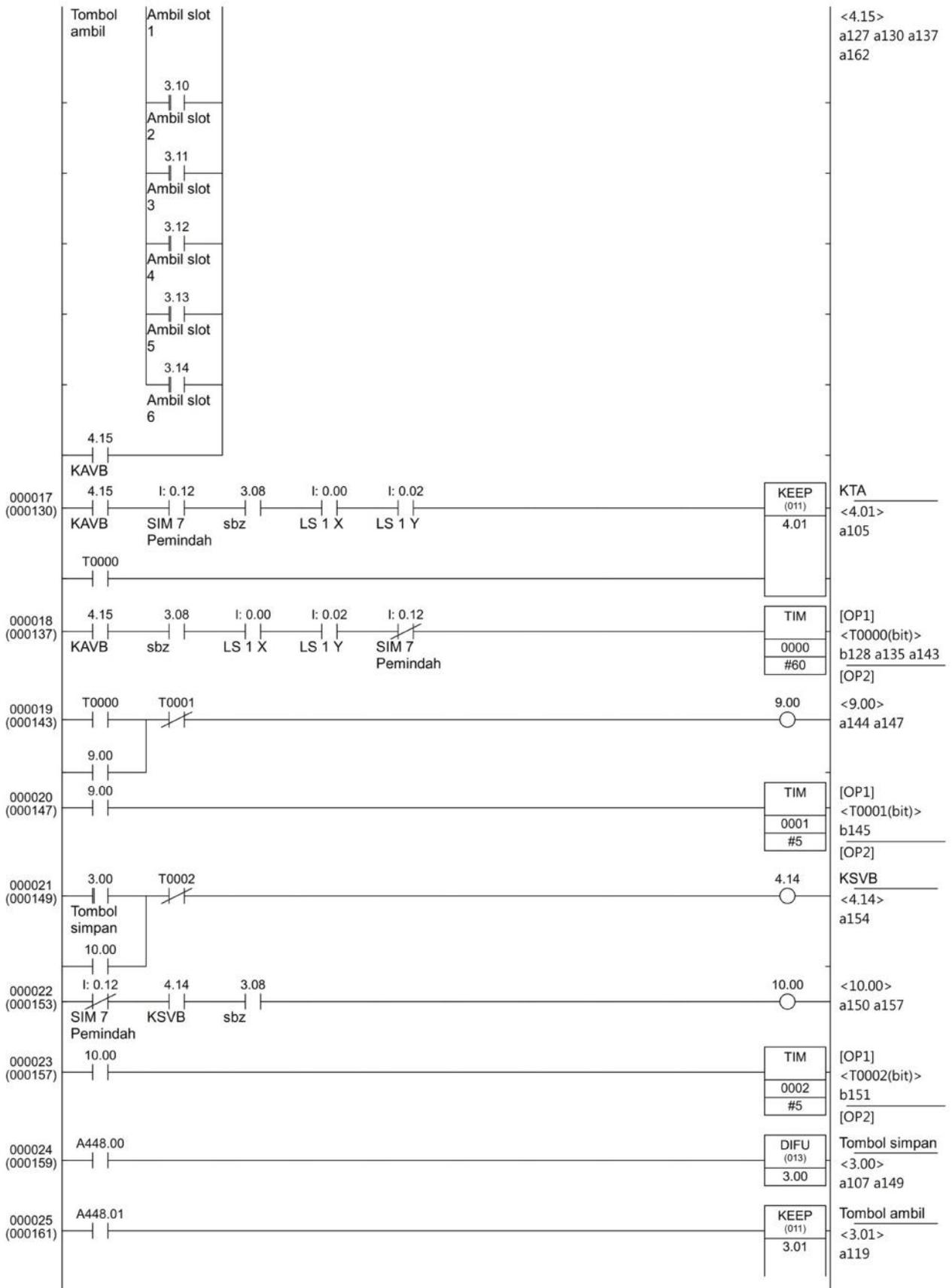
```

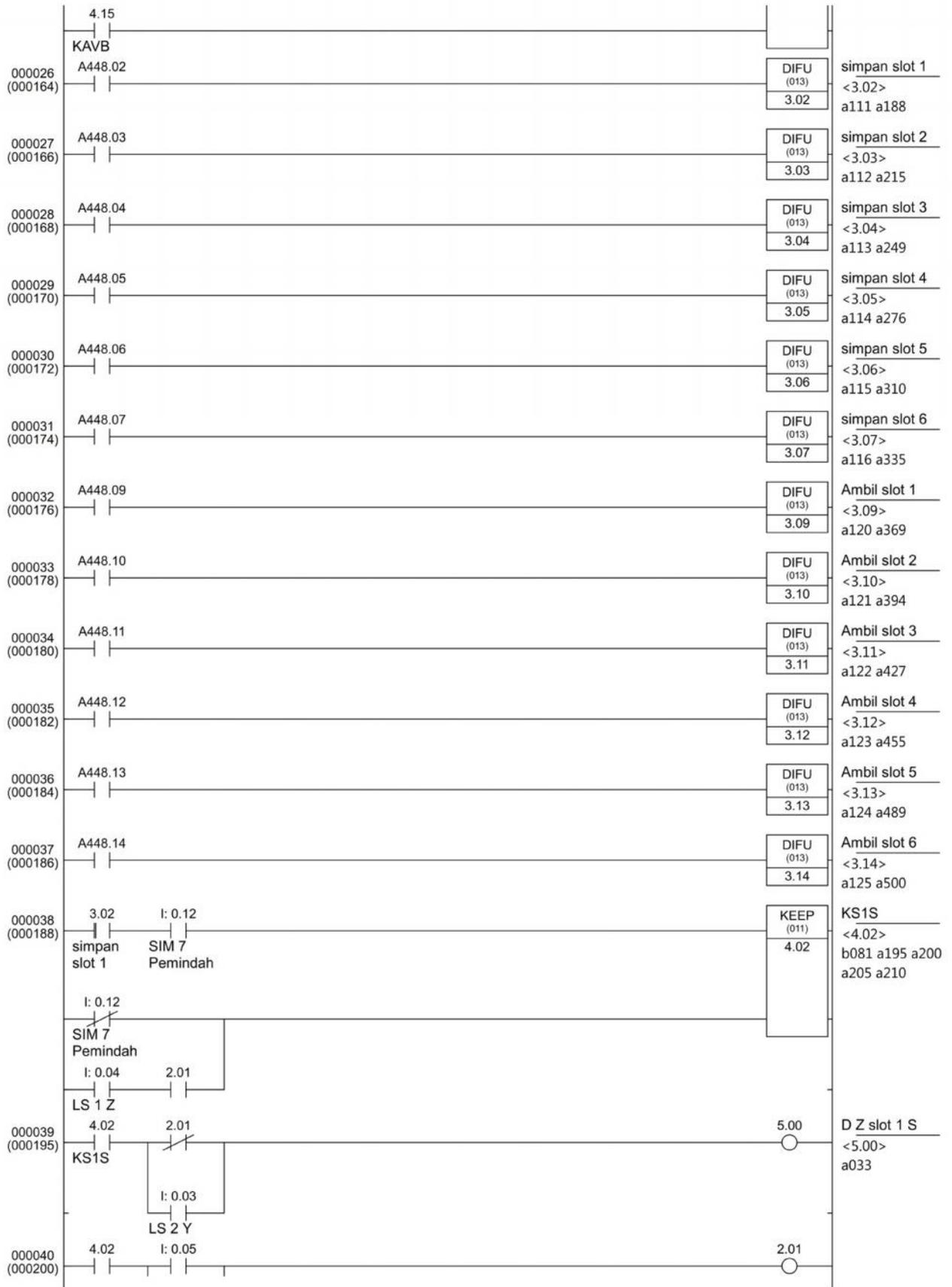


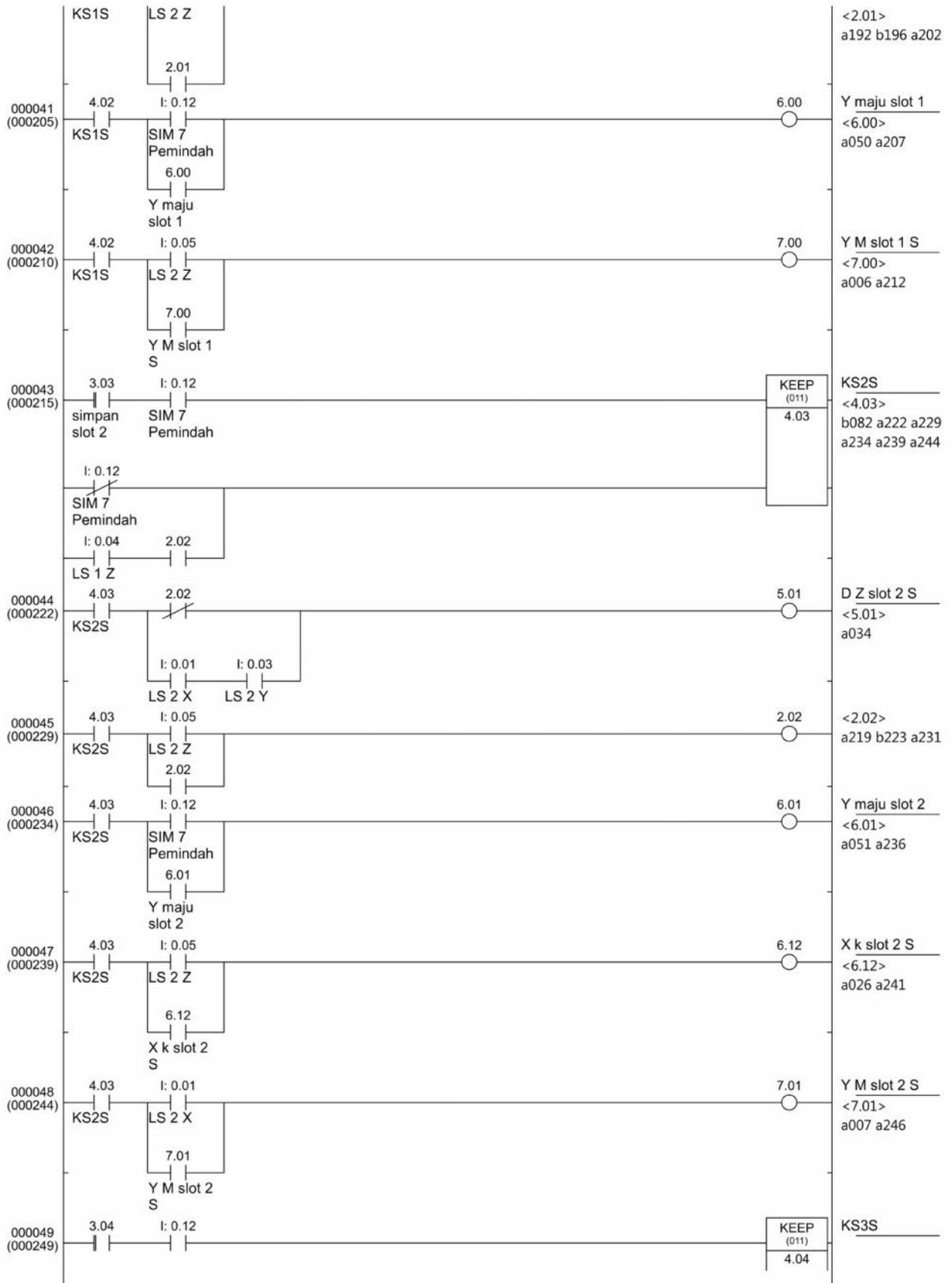


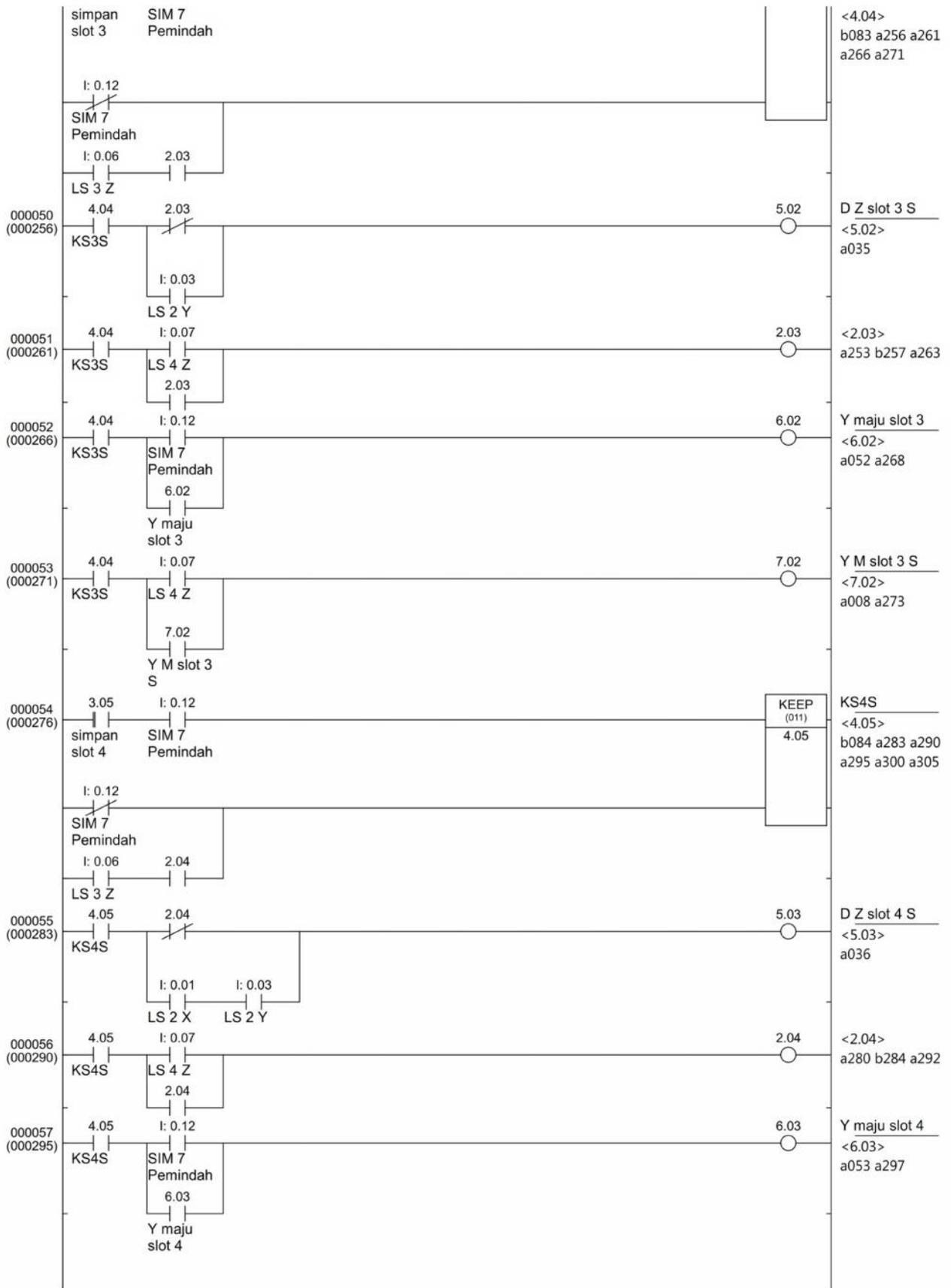


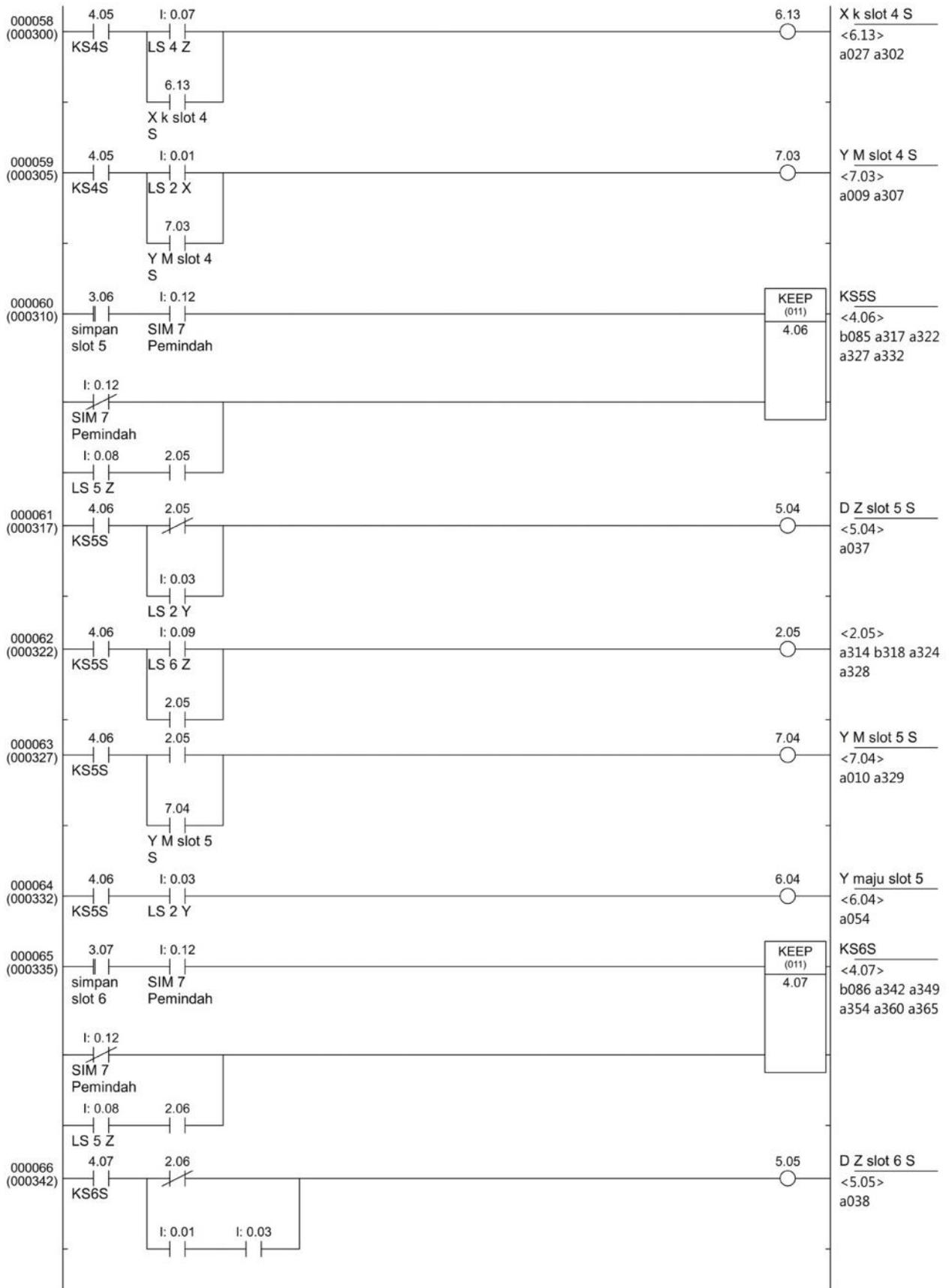


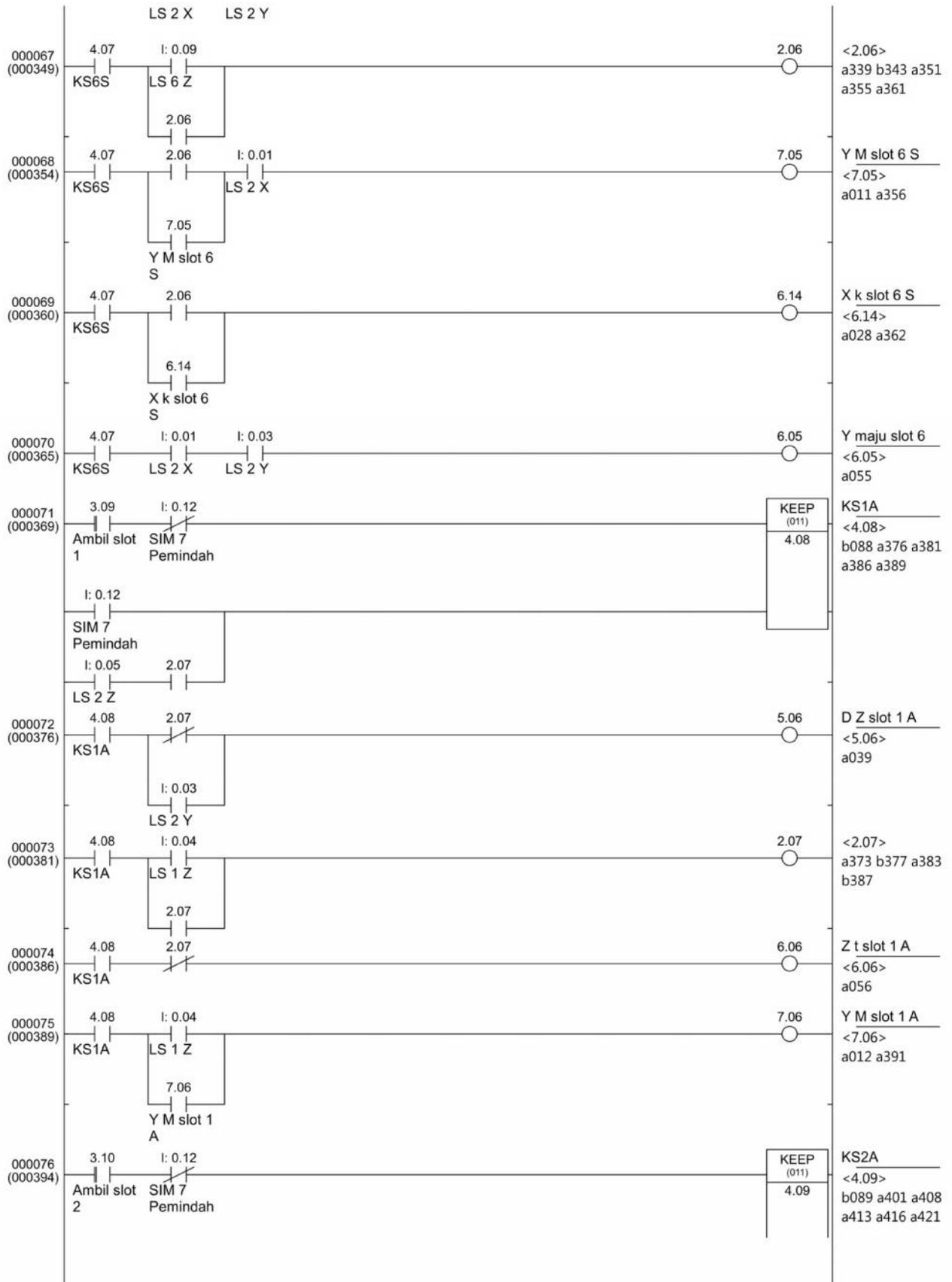


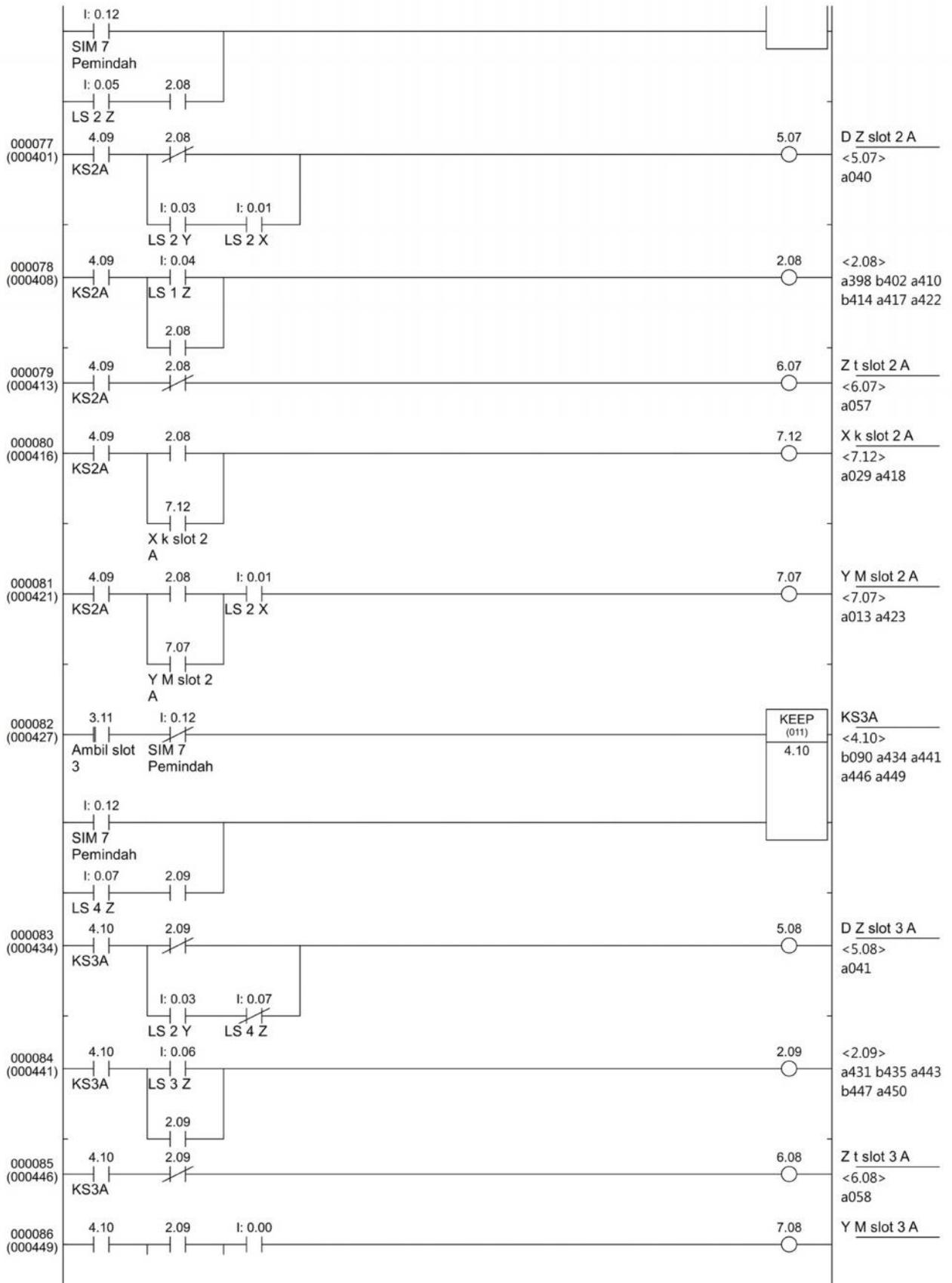


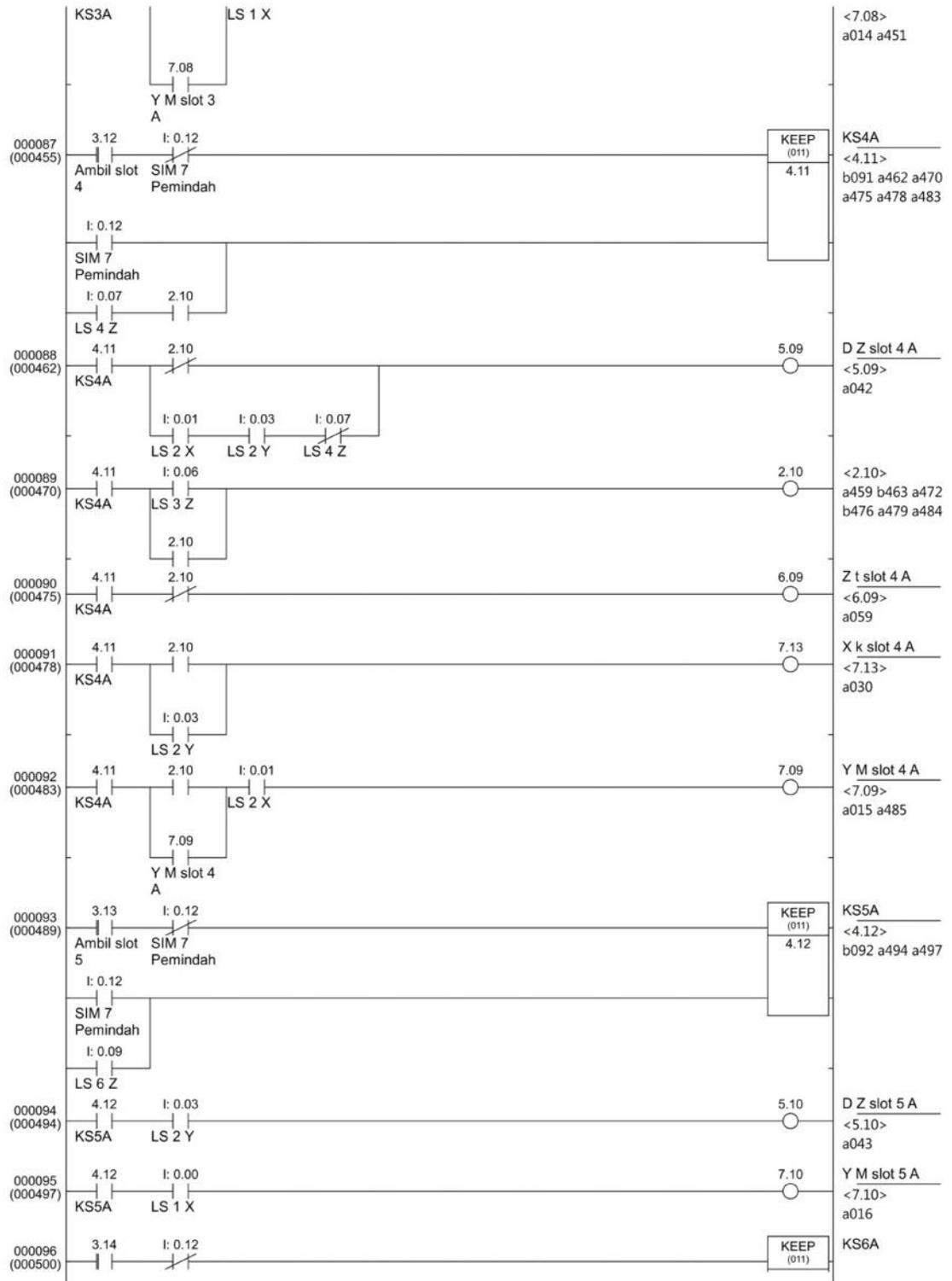


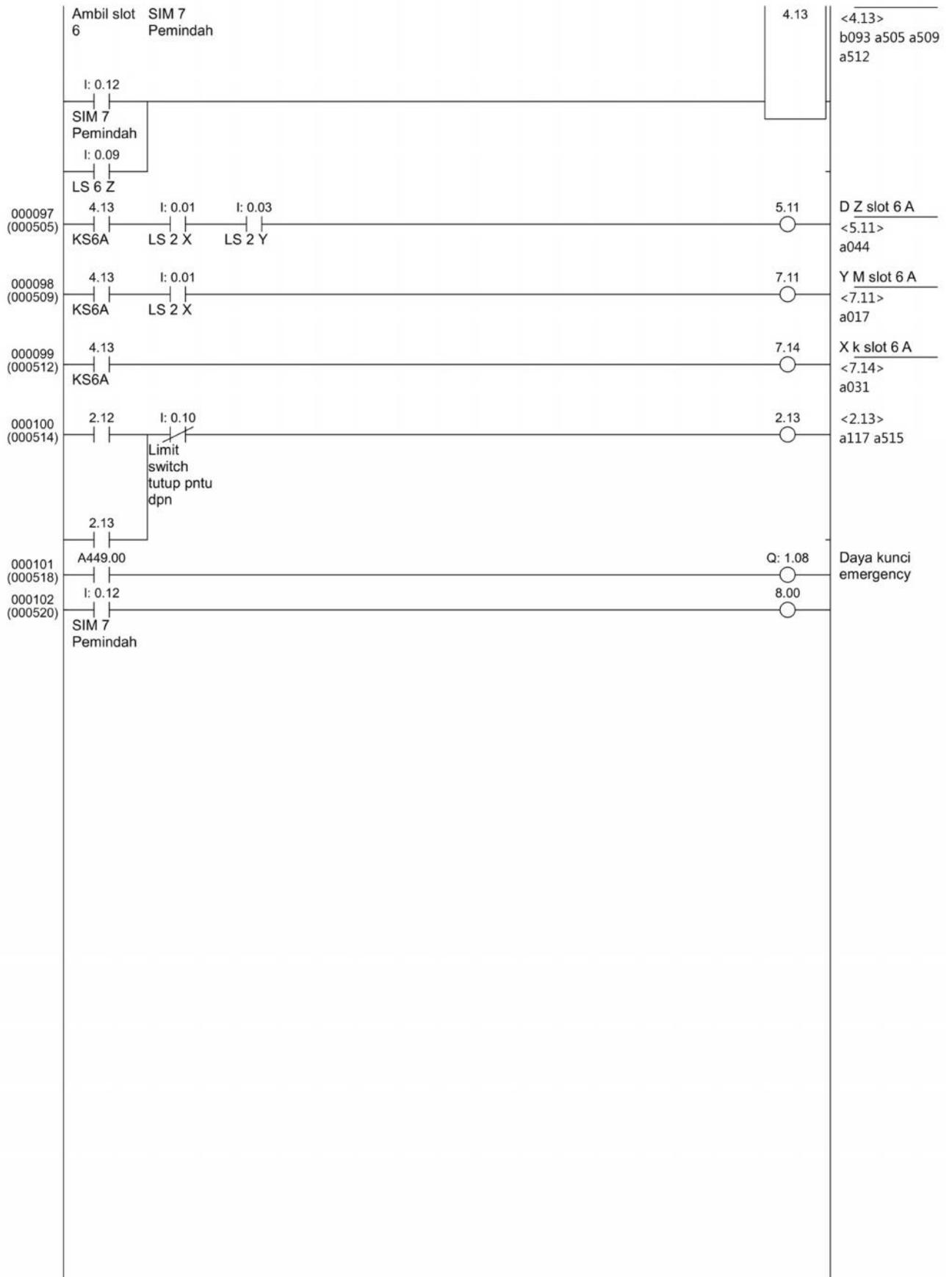




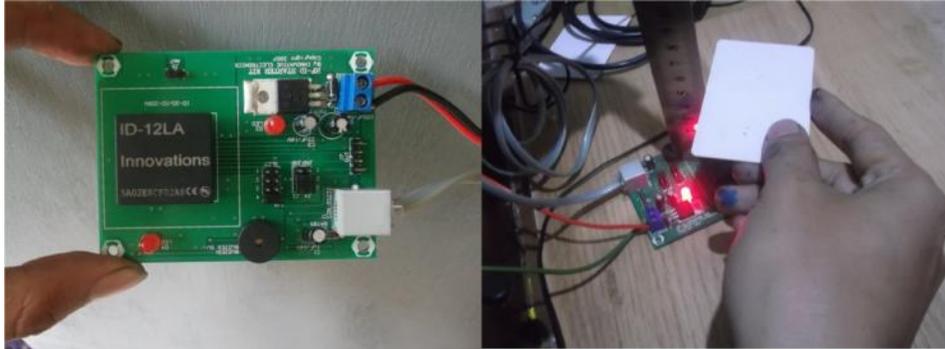




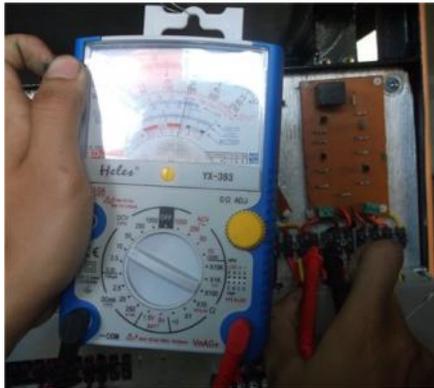




Lampiran Foto



RFID KIT dan Pengujiannya



Pengukuran Seonsor Infra merah



Motor Penggerak *Prototype*



Door Lock



Pemindah Benda



Tampak Samping



Tampak Depan

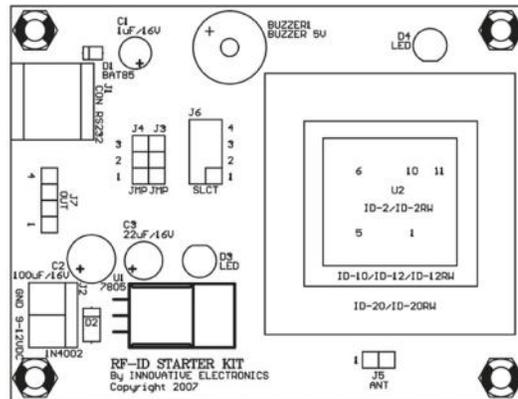
RFID Starter Kit

RFID Starter Kit merupakan suatu sarana pengembangan RFID berbasis reader tipe ID-12 yang telah dilengkapi dengan jalur komunikasi RS-232 serta indikator buzzer dan LED. Modul ini dapat digunakan dalam aplikasi mesin absensi RFID, RFID access controller, dsb.

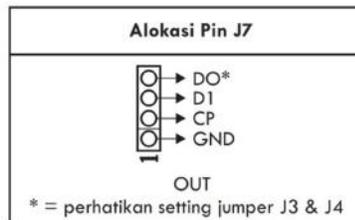
Spesifikasi

1. Berbasis RFID reader ID-12 dengan frekuensi kerja 125 kHz untuk kartu berformat EM4001/sejenis dan memiliki jarak baca maksimal 12 cm.
2. Kompatibel dengan varian RFID reader lainnya, antara lain: ID-2, ID-10, dan ID-20.
3. Mendukung varian RFID reader/writer, antara lain: ID-2RW, ID-12RW, dan ID-20RW.
4. Mendukung format data ASCII (UART TTL/RS-232), Wiegand26, maupun Magnetic ABA Track2 (Magnet Emulation).
5. Dilengkapi dengan buzzer sebagai indikator baca, serta LED sebagai indikator tulis.
6. Tersedia jalur komunikasi serial UART RS-232 dengan konektor RJ11.
7. Tegangan input catu daya 9 - 12 VDC (J2).

Tata Letak dan Setting Jumper



Antena dapat dihubungkan ke J5 untuk RFID reader only atau reader/writer yang memerlukan antena eksternal, seperti ID-2 dan ID-2RW.



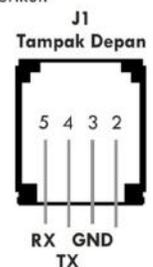
J7 hanya digunakan untuk RFID reader only dalam mode UART TTL (ASCII), Wiegand26, dan Magnet Emulation. J7 tidak boleh digunakan pada mode lain.

Pengaturan jumper J3, J4, dan J6 harus disesuaikan dengan jenis RFID (reader only atau reader/writer) serta format data RFID reader yang akan digunakan.

Setting Jumper J3, J4, & J6	
	RFID reader only dengan format data UART RS-232 (ASCII).
	RFID reader only dengan format data UART TTL (ASCII).
	RFID reader only dengan format data Wiegand26
	RFID reader only dengan format data Magnet Emulation
	RFID reader/writer dengan antarmuka UART RS-232.

Adapun hubungan antara komputer dengan RFID Starter Kit adalah "Straight" dengan konfigurasi sebagai berikut:

COM port Komputer DB9	RFID Starter Kit J1
RX (pin 2)	RX (pin 5)
TX (pin 3)	TX (pin 4)
GND (pin 5)	GND (pin 3)



J1 hanya digunakan untuk RFID reader only dalam mode UART RS-232 (ASCII) dan RFID reader/writer. Pada mode lain, J1 tidak boleh digunakan dan kabel tidak boleh terhubung.

Isi CD

1. Contoh Aplikasi dan Program Testing.
2. Datasheet RFID Reader ID-12.
3. Manual RFID Starter Kit.
4. Website Innovative Electronics

DAFTAR RIWAYAT HIDUP



Muhammad Rizki Athoillah dilahirkan pada tanggal 12 November 1993 di Majalengka, dari pasangan Bapak Dedy Dwidjaya dan Ibu Jojoh Johariyah sebagai anak pertama dari dua bersaudara. Memiliki nama panggilan Rizki. Pendidikan yang ditempuh penulis adalah di R.A. Al-Badariyah Cibungbulang Bogor tahun 1998-1999, SD Negeri Cibatok IV Cibungbulang Bogor pada tahun 1999-2005, SMP Negeri 1 Cibungbulang Bogor tahun 2005-2008, SMA Negeri 1 Leuwiliang tahun 2008-2011. Ketika kelas VI SD penulis menjadi siswa lulusan terbaik di SD Negeri Cibatok IV. Ketika belajar di SMA penulis sering mengikuti kepanitiaan acara-acara besar seperti perlombaan antar generasi islam se-Bogor raya pada tahun 2009 dengan menjabat sebagai ketua bidang publikasi dekorasi dan dokumentasi, dan kajian islami pelajar dan pemuda se-Bogor raya bekerja sama dengan Hasmi Leuwiliang pada tahun 2011 dengan menjabat sebagai ketua bidang publikasi dekorasi dan dokumentasi. Pada tahun 2011 Penulis melanjutkan pendidikan ke Universitas Negeri Jakarta, Rawamangun, Jakarta Timur, melalui Jalur UMB (Ujian Masuk Bersama) dan diterima di jurusan Teknik Elektro Universitas Negeri Jakarta, Program Studi Pendidikan Teknik Elektro. Pada Semester kedua menempuh pendidikan di Universitas Negeri Jakarta, penulis pernah mengikuti pelatihan kepemimpinan mahasiswa jurusan Teknik Elektro.