

PERBAIKAN DESAIN RLL (*RELAY LADDER LOGIC*) PENGENDALIAN MOTOR PENGGERAK PINTU SUMBER RADIOAKTIF PADA PEREKAYASAAN PORTAL MONITOR

Donny Nurmayady, Dian Fitri Atmoko
Pusat Rekayasa Perangkat Nuklir (PRPN) – BATAN
E-mail : donny@batan.go.id

ABSTRAK

PERBAIKAN DESAIN RLL (RELAY LADDER LOGIC) PENGENDALIAN MOTOR PENGGERAK PINTU SUMBER RADIOAKTIF PADA PEREKAYASAAN PORTAL MONITOR. Penutup sumber radioaktif pada kegiatan perekayasa portal monitor di Pusat Rekayasa Perangkat Nuklir (PRPN) digerakkan dengan menggunakan mesin listrik (motor). Pergerakan motor dikendalikan oleh Programmable Logic controller (PLC) melalui bahasa pemrograman RLL. Pemrograman RLL telah dirancang untuk pergerakan motor penutup sumber. Dilakukan evaluasi keberurutan proses (keadaan) terhadap desain pemrograman RLL ini. Hasil evaluasi menunjukkan masih ada kemungkinan proses (keadaan) yang saling mendahului dan bekerja tidak sesuai tahapan perekayasa yang diinginkan. Ketidak berurutan proses ini dapat disebabkan oleh masuknya inputan yang tidak diharapkan dikarenakan kerusakan fisik peralatan input ataupun kesengajaan. Pemrograman RLL seyogyanya mampu menjaga keberurutan proses sistem secara perangkat lunak. Makalah ini membahas mengenai evaluasi desain RLL yang sudah ada dan dilakukan perbaikan desain terhadap desain tersebut. Selanjutnya dilakukan evaluasi terhadap desain perbaikan tersebut. Evaluasi desain pada makalah ini menggunakan metode state diagram. Desain perbaikan memberikan hasil keberurutan (sekuensial) proses (keadaan) sesuai dengan tahapan kegiatan yang diharapkan.

Kata kunci: RLL, Keberurutan, mesin listrik, sumber radioaktif

ABSTRACT

DESIGN IMPROVEMENT OF RLL (RELAY LADDER LOGIC) MOTOR CONTROL FOR A RADIOACTIVE SOURCE GATE ON PORTAL MONITOR DEVELOPMENT. Radioactive source gate on Portal Monitor development in Centre of Nuclear Components Engineer is moved by electric machine (motor). The movement of this motor is controlled by Programmable Logic Controller (PLC) on RLL language program. RLL program had been designed for a radioactive source gate. This design has been evaluated and presented some program lists can precede each other and produced non sequential events. This non sequential events caused by some undesirable inputs enter to the proses. This undesirable input was activated by physical error and human error. RLL programs should maintain the process in sequence on software base. In this paper, RLL program will be evaluated by state diagram method's. RLL program will be re-designed. Design improvement results a sequential event in accordance with stages.

Keywords: RLL, Sequential, electric machine, radioactive source

1. PENDAHULUAN

Kegiatan kerekayasa portal monitor di Pusat Rekayasa Perangkat Nuklir mempunyai misi melakukan verifikasi dan validasi isi petikemas. Verifikasi dan validasi merupakan persyaratan utama ekspor/impor barang antar negara. Perekayasa portal monitor merupakan verifikasi dan validasi secara otomatis, dilakukan dengan teknik sistem pencitraan. Salah satu sistem pencitraan yang dapat digunakan untuk memindai petikemas adalah dengan teknik transmisi serapan sinar gamma. Sistem ini terdiri dari sumber sinar gamma Co^{60} atau Cs^{137} dan *line scan camera*^[1]. Perakayasa Portal Monitor dirancang untuk dapat mendeteksi keberadaan “sesuatu” yang berada di dalam petikemas^[1].

Seluruh sumber radiasi gamma Co^{60} atau Cs^{137} pada penelitian ini telah dikungkung oleh bahan pelindung sesuai dengan perhitungan dan ketentuan yang berlaku. Namun pada saat akan digunakan, sumber tersebut akan dibuka dengan bukaan (diameter) tertentu untuk digunakan. Pembukaan ini akan menyebabkan terjadi paparan sinar gamma ke lingkungan. Sinar gamma merupakan sumber radiasi memiliki energi yang cukup besar untuk menginduksi reaksi ionisasi saat berinteraksi dengan materi sehingga terjadi pelepasan elektron dari atom-atom penyusun materi yang dilintasinya. Interaksi sinar gamma dengan tubuh manusia dengan energi melampaui batas dan berlangsung lama akan menyebabkan kerusakan DNA manusia^[2]. Prinsip ALARA dalam penanganan radioaktif mutlak dilaksanakan^[3]. Pembukaan kontainer/kungkung sumber radioaktif secara langsung oleh seorang pengguna akan membuat pengguna tersebut terpapar radioaktif. Efek paparan radioaktif dapat dirasakan secara stokastik maupun scholastic. Kedua efek tersebut sama bahayanya. Diperlukan perangkat pembantu pembuka kontainer sumber radioaktif agar pengguna tidak secara langsung membuka container. Pembuka kontainer radioaktif meskipun berdimensi kecil tetapi memiliki massa yang besar. Perangkat pembuka ini harus mampu membuka/mengangkat pembuka container. Perangkat ini hendaknya mampu diaktifkan dari jarak jauh melalui pengendalian sebuah personal komputer. Hendaknya perangkat ini pula bekerja secara otomatis sesuai unjuk kerja yang diinginkan. Dengan mempertimbangkan kondisi tersebut dibuatlah pembukaan penutup radiasi menggunakan motor listrik dan dikendalikan dari jarak jauh dengan menggunakan bantuan komputer.

Programmable Logic Controller (PLC) telah lama dikenal sebagai perangkat pengendalian menggunakan komputer dan dapat dikendalikan dari jarak jauh^[4]. Bahasa pemrograman di PLC yang biasa dikenal dengan *ladder logic* merupakan bahasa pemrograman yang mudah dipahami. Dalam beberapa literatur *ladder logic* diagram disebut juga *Relay Ladder Logic* (RLL)^{[5][6]}.

Telah dilakukan pembuatan desain RLL. Desain ini telah dijalankan dan memberikan hasil sesuai dengan tahapan kegiatan operasinya. Meskipun demikian desain ini terlihat masih belum sempurna, Dimungkinkan terjadinya proses yang tidak sekuensial (berurutan) dimana satu proses dapat melompat ke proses lain yang tidak sesuai dengan tahapan yang diinginkan. Lompatan ini dapat terjadi ketika adanya kerusakan (*error*) pada masukan ataupun keluaran PLC baik disengaja maupun tidak. Kerusakan ini dapat mengakibatkan proses yang saling tumpang tindih dan saling mendahului pada masing-masing tahapan. Selain itu catu daya motor penggerak tutup sumber radioaktif menggunakan akumulator terpisah. Kemungkinan error juga dapat terjadi disini. Ada kemungkinan motor tidak berfungsi optimal dikarenakan kerusakan pada catu dayanya. Kondisi ini memerlukan bantuan operator untuk memeriksa ke lokasi motor ini. Dan diperlukan indikator yang memberi tahu kerusakan ini.

Ada beberapa cara analisis yang dilakukan untuk mendeteksi kesalahan pemrograman RLL^{[7][8]}. Analisis kesalahan pemrograman RLL dengan mengubah bahasa pemrograman RLL menjadi bahasa pemrograman tingkat tinggi dalam bentuk *state-chart* dan dituangkan dalam bentuk prototipe virtual^[9]. Penunjukan kesalahan RLL dilakukan dengan pendeteksian secara otomatis dengan menambahkan program tambahan yang akan mendeteksi kesalahan yang umum dan fatal biasa terjadi dalam pemrograman RLL^[6]. Telah dilakukan penelitian analisis deteksi kesalahan pemrograman RLL berbasis pada batasan-batasan tertentu^[8]. Batasan inilah yang akan dideteksi menjadi suatu informasi untuk pendeteksian kesalahan pemrograman RLL. Dalam verifikasi RLL dilakukan dengan menggunakan bantuan alat Candance SMV dimana alat ini akan memeriksa simbol RLL, dari pemeriksaan simbol RLL diharapkan tidak ada kesalahan yang terjadi pada pemrograman RLL^[10]. Selain itu penunjukan bahwa analisis kesalahan RLL dilakukan dapat dilakukan dengan menggunakan metode petrinet dengan bantuan alat FIACRE Tool^[11]. Kesemua metode di atas sangat tepat dilakukan pendeteksian kesalahan RLL pada saat waktu-nyata. Karena metode analisis pendeteksian dapat diprogram di luar keadaan PLC. Selain itu metode-metode di atas biasanya digunakan untuk pemrograman RLL yang sudah kompleks dan banyak serta sulit untuk dideteksi

secara visual, namun demikian metode–metode tersebut memerlukan pendekatan matematika logika yang mendalam. Analisis kesalahan juga dapat dilakukan dengan cara menjalankan langsung program RLL. Kemudian dianalisis dengan menggunakan metode *timing diagram*^{[12][13]}. Untuk keberurutan (sekuensial) suatu program dapat juga dilakukan dengan menggunakan metode *state diagram*^[15]. Metode menjalankan langsung ini sangat rentan untuk sistem yang besar dan kompleks. Kerentanan ini juga akan berdampak pada kesalahan yang fatal dan berbahaya. Namun untuk sistem yang kecil dan tidak memerlukan pengetahuan matematika logika yang kompleks metode ini dapat dijadikan suatu alternatif. Bahkan untuk RLL yang relatif kecil dapat dianalisis terlebih dahulu dan diperbaiki jika ada kesalahan untuk kemudian dijalankan program RLL nya.

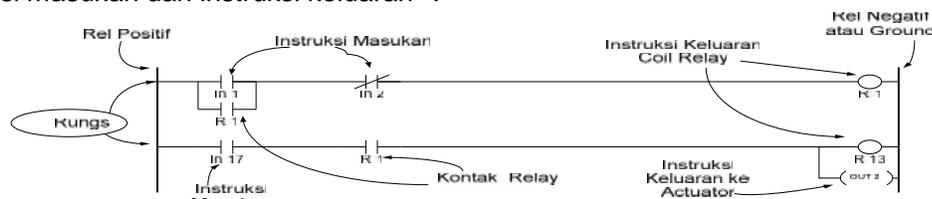
Dalam makalah ini akan dibahas mengenai evaluasi desain RLL yang sudah ada dan akan digunakan analisis kesalahan dengan menggunakan metode *state diagram*. Meskipun penelitian ini terdapat unsur-unsur radioaktif, namun sistem pengendalian masih sangat sederhana. Untuk selanjutnya akan dilakukan perbaikan desain RLL. Dengan metode analisis kesalahan yang sama, desain yang baru ini juga akan dievaluasi. Desain baru ini diharapkan dapat menekankan pada aspek keberurutan proses secara pengkabelan maupun secara pemrograman, pengawasan terhadap kemungkinan motor bermasalah dan aspek keselamatan. Pemrograman RLL ini dibatasi pada pengendalian motor pembuka-penutup sumber radioaktif. Penamaan pada pemrograman mengikuti format yang telah ada.

1.1. TEORI

1.1.1 Programable Logic Controller (PLC)

PLC merupakan sebuah pemrograman, yang terdiri dari sebuah mikroprosesor.^[5] PLC digunakan untuk mengendalikan sebuah sistem secara otomatis. Kelebihan dari PLC dibanding dengan pengendalian otomatis elektro mekanik adalah penggunaan *relay* maya untuk otomatisasi dan diagram rangkaian logika *relay* dibentuk dalam bentuk tangga. PLC menggantikan *relay* elektromekanik menjadi *relay* imajiner yang tertanam di dalam program PLC. PLC mampu menghubungkan dan mengendalikan secara fisik masukan berupa sensor atau saklar ke keluaran suatu *actuator* berupa perintah *on/off* kontak *relay*. Ada beberapa metode pemrograman PLC, salah satunya dengan diagram logika tangga atau *ladder logic diagram*. Pada penelitian ini *ladder logic diagram* disebut juga *Relay Ladder logic (RLL)*^[5].

RLL adalah sebuah bahasa pemrograman yang telah digunakan secara luas untuk aplikasi kendali yang kompleks dan tertanam. Pemrograman RLL dihadirkan dalam bentuk diagram tangga (*ladder*), yang menggambarkan suatu sirkit atau pun diagram alir kerja. RLL terdiri dari satu set anak-anak tangga (*rung*), dimana setiap *rung* memiliki instruksi masukan dan instruksi keluaran^[6].



Gambar 1. Contoh RLL

Pada gambar 1, adalah sebuah contoh diagram RLL yang sangat sederhana dimana terdapat dua buah *rung* yang terhubung pada rel kutub positif dan rel kutub negatif atau rel *ground*. Garis kecil menurun | | and || menggambarkan instruksi masukan.

Pemrograman RLL pertama kali akan membaca perintah masukan dari instruksi masukan eksternal dan menjalankan instruksi keluaran dari *rung* secara sekuensial dari kiri ke kanan dan dari atas ke bawah *ladder*^[9]. Meskipun RLL telah luas dikenal dan mudah dimengerti, tetapi pemrograman RLL ini kerap diperlukan ketelitian, terlebih lagi

ketelitian dalam hal keberurutan proses perjalanan (sekuensial). Kesalahan dalam pemrograman RLL akan berdampak pada terganggunya proses-proses pada sistem, akan lebih bahaya lagi jika proses tersebut menyangkut keselamatan manusia akan terjadi kejadian yang membahayakan jiwa manusia^[10].

Pada makalah ini akan digunakan analisis kesalahan dengan menggunakan metode *state diagram* dan hasilnya akan ditampilkan ke dalam *timing diagram*. Meskipun penelitian ini terdapat unsur-unsur radioaktif, namun sistem pengendalian masih sangat sederhana. Penggunaan *timing diagram* telah lama dikenal dalam aplikasi elektronika logika. Salah satu untuk mengetahui kesalahan pada program RLL adalah dengan menjalankan secara keseluruhan kemudian dianalisis dengan menggunakan *state diagram* dan hasilnya di gambarkan ke dalam *timing diagram*.

1.1.2. State Diagram (Diagram Keadaan)

Sebuah diagram keadaan (*state diagram*) adalah jenis diagram yang digunakan dalam ilmu komputer dan bidang terkait untuk menggambarkan perilaku sistem. Diagram keadaan mengharuskan sistem yang dijelaskan terdiri dari sejumlah keadaan, bahkan dapat dijadikan sebuah pendahuluan. Diagram keadaan digunakan untuk memberikan deskripsi ringkasan abstraksi tentang perilaku dari suatu sistem. Perilaku ini dianalisis dan direpresentasikan dalam suatu peristiwa secara seri, yang dapat terjadi dalam satu atau lebih kemungkinan keadaan. Dengan ini setiap diagram biasanya mewakili objek dari satu kelas dan menghubungkan dengan keadaan berbeda yang berbeda dari objek melalui sistem. Perubahan dari satu keadaan ke keadaan lain ketika diprakarsai oleh suatu peristiwa atau kondisi yang memicu. Perubahan ini disebut transisi^[16]. Perilaku dari sebuah diagram keadaan dapat dilihat di berbagai perangkat di dalam masyarakat modern yang bekerja secara berurutan (sekuensial) sesuai dengan tahapan yang ditelaah ditentukan.^[17]

Sebuah contoh sederhana dari suatu diagram keadaan adalah mekanisme pintu putar pada bus transjakarta. Contoh ini prinsip kerjanya hampir sama dengan [18], [19]. Sebuah pintu putar digunakan untuk mengakses masuk ke dalam shelter bus transjakarta, adalah sebuah gerbang (pintu) dengan tiga lengan berputar pada ketinggian pinggang yang terletak pada pintu masuk shelter setelah tempat pembelian tiket. Pintu pagar memiliki dua keadaan: terkunci dan tidak terkunci. Ada dua masukan yang mempengaruhi keadaannya : masukan kartu ke dalam lubang dan mendorong lengan pintu.

Diagram keadaan pintu putar dapat disajikan dalam bentuk diagram keadaan beserta tabel transisinya (gambar 2) yang memperlihatkan masing-masing keadaan, keadaan baru dan hasil (keluarannya/aksinya) yang dihasilkan dari input tersebut



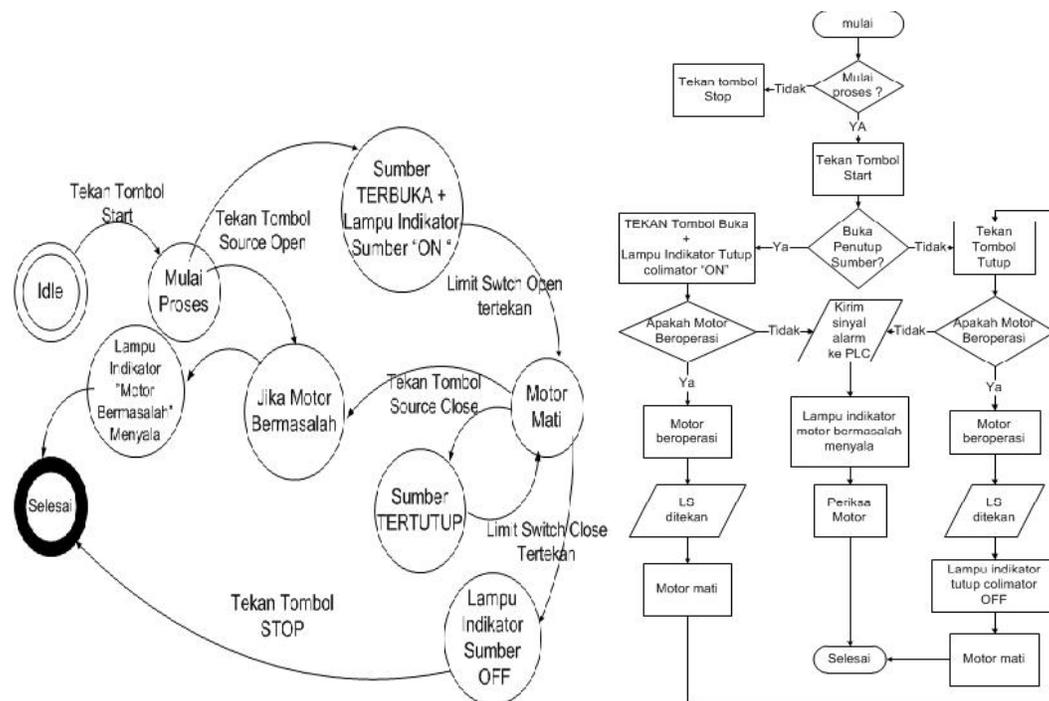
Gambar 2. Contoh Diagram keadaan dan Tabel Transisinya^[18,19,20]

Pada diagram keadaan setiap keadaan diwakili oleh lingkaran (*node*). Arah panah menunjukkan transisi dari satu keadaan menuju keadaan yang lain. Setiap panah diberi label masukan yang menuju keadaan yang baru tersebut. Masukan yang menyebabkan tidak adanya perubahan (seperti masukan kartu ke dalam pada status tidak terkunci)

diwakili oleh panah melingkar kembali ke keadaan semula. Lingkaran ganda (lingkaran keadaan terkunci) menunjukkan awal proses.

2. METODOLOGI

Tahapan kegiatan dimulai dengan menekan tombol (masukan) *start*. Masukan ini bersifat *interlock* dengan masukan *stop*. Kemudian memberi masukan *Source Open*. Pada tahap ini tutup sumber radioaktif akan bergerak membuka. Gerakan ini akan dilakukan dengan bantuan motor listrik DC. Gerakan motor ini akan berhenti ketika *limit switch open* diberi masukan (tertekan). Sumber terbuka dan lampu indikator sumber menyala. Kemudian diberikan masukan *Source Close*. Pada tahap ini tutup sumber radioaktif akan bergerak menutup. Gerakan ini akan dilakukan dengan bantuan motor listrik DC. Gerakan atau motor ini akan berhenti ketika *limit switch close* diberi masukan (tertekan). Sumber tertutup dan lampu indikator sumber mati. Motor listrik DC sebagai penggerak mendapat catuan terpisah dari *accumulator*. Ada kemungkinan motor tidak dapat beroperasi optimal (tidak dapat bergerak). Dengan demikian terjadi kerusakan proses dan membutuhkan teknisi untuk datang ke lokasi untuk memeriksa peralatan. Akan ada sinyal yang aktif (dari *relay*) ketika catu daya motor mengalami gangguan. Sinyal ini akan memberikan masukan ke PLC dan akan memberikan keluaran berupa lampu indikator motor bermasalah di lokasi. Jika semua kegiatan sudah selesai maka ditekan tombol *stop*. Tahapan kegiatan ini seperti ditampilkan pada gambar 4.

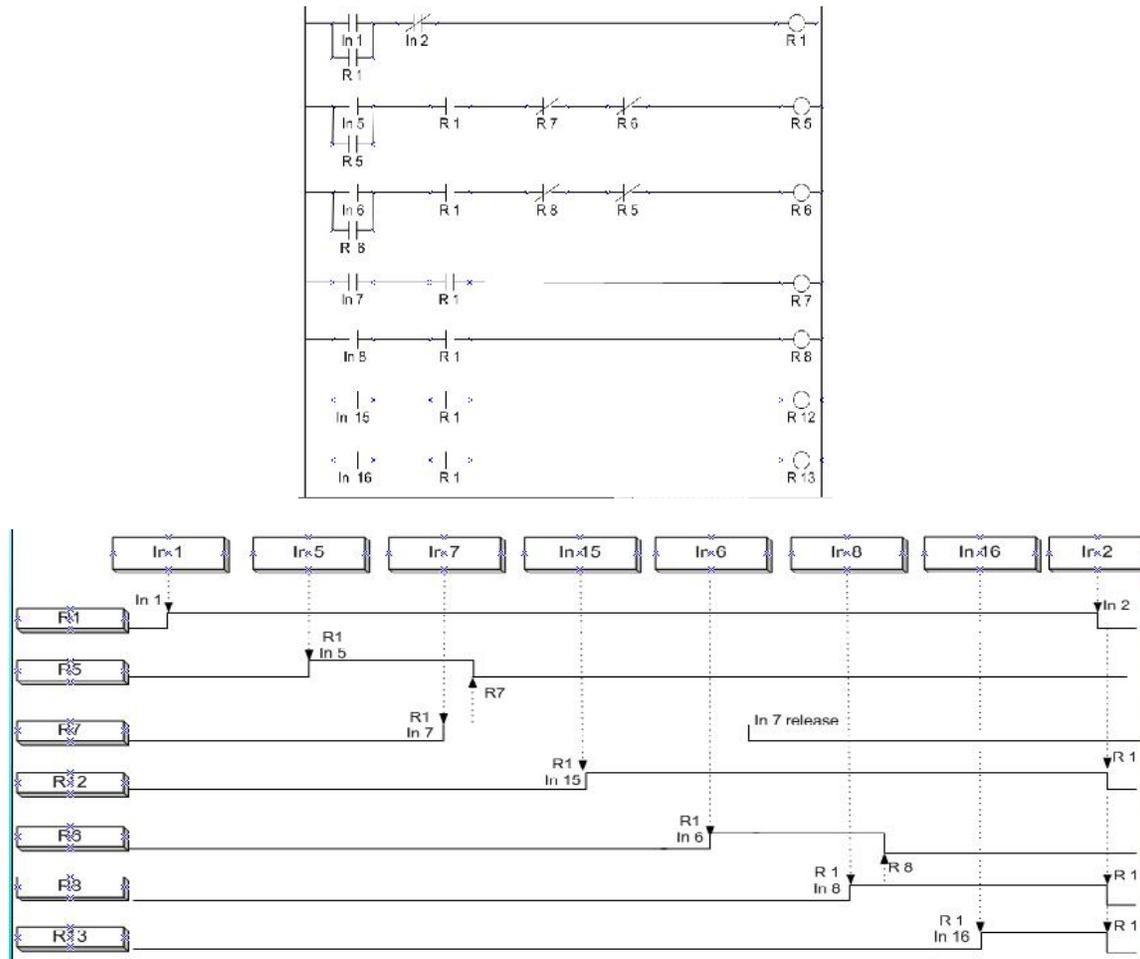


Gambar 4. Tahapan Kegiatan dan diagram alir kegiatan

Setelah diketahui tahapan kegiatan yang diinginkan selanjutnya adalah pemrograman pengendalian menggunakan PLC dengan menuangkan ke dalam metode RLL. Pemrograman RLL pada kegiatan ini telah dilakukan dan telah diuji, namun demikian masih terlihat belum sempurna^{[20][21]}. Meskipun terbukti berhasil namun unsur keberurutan (sekuensial) tahapan yang dituangkan dalam pemrograman RLL PLC masih ada kekurangan^[10]. Sehingga masih mungkin terjadi kesalahan yang diakibatkan kelalaian manusia ataupun penyebab lainnya. Pada program RLL terdahulu belum adanya kemungkinan kehilangan catu daya motor^{[20][21]}. Kemungkinan kehilangan catu daya motor

sangat berbahaya bagi lingkungan maupun terutama ketika sumber sedang dalam posisi terbuka. Oleh karena itu dilakukan desain ulang pemrograman RLL.

Di bawah ini gambar 5 merupakan desain pemrograman RLL yang sudah ada^[21]

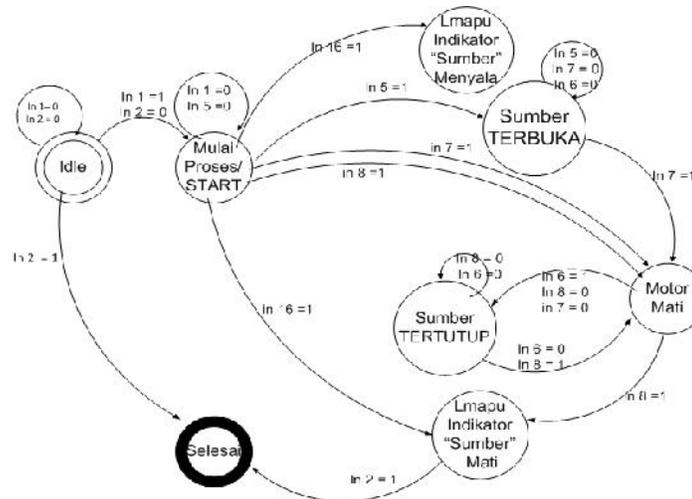


Gambar 5. RLL desain A^[22] dan diagram waktunya

Pada diagram waktu gambar 5 menunjukkan hasil proses kerja dari desain A. Diagram waktu ini menunjukkan output R13 (indikator *Source Open*) berhenti dikarenakan tombol *stop*. Bukan dikarenakan proses setelahnya (tekan tombol tutup). Meskipun secara pengkabelan masukan In 15 berasal dari umpan balik keluaran R 5 yang disambung ke In 15.

3. HASIL DAN PEMBAHASAN

Dari desain A gambar 5. akan dituangkan ke dalam diagram *state machine*. "Diagram *state machine* digunakan sebagai alat analisis untuk melihat konsistensi hasil disain"^[15]. Di bawah ini gambar 6 adalah diagram *state machine* dari pemrograman RLL desain A

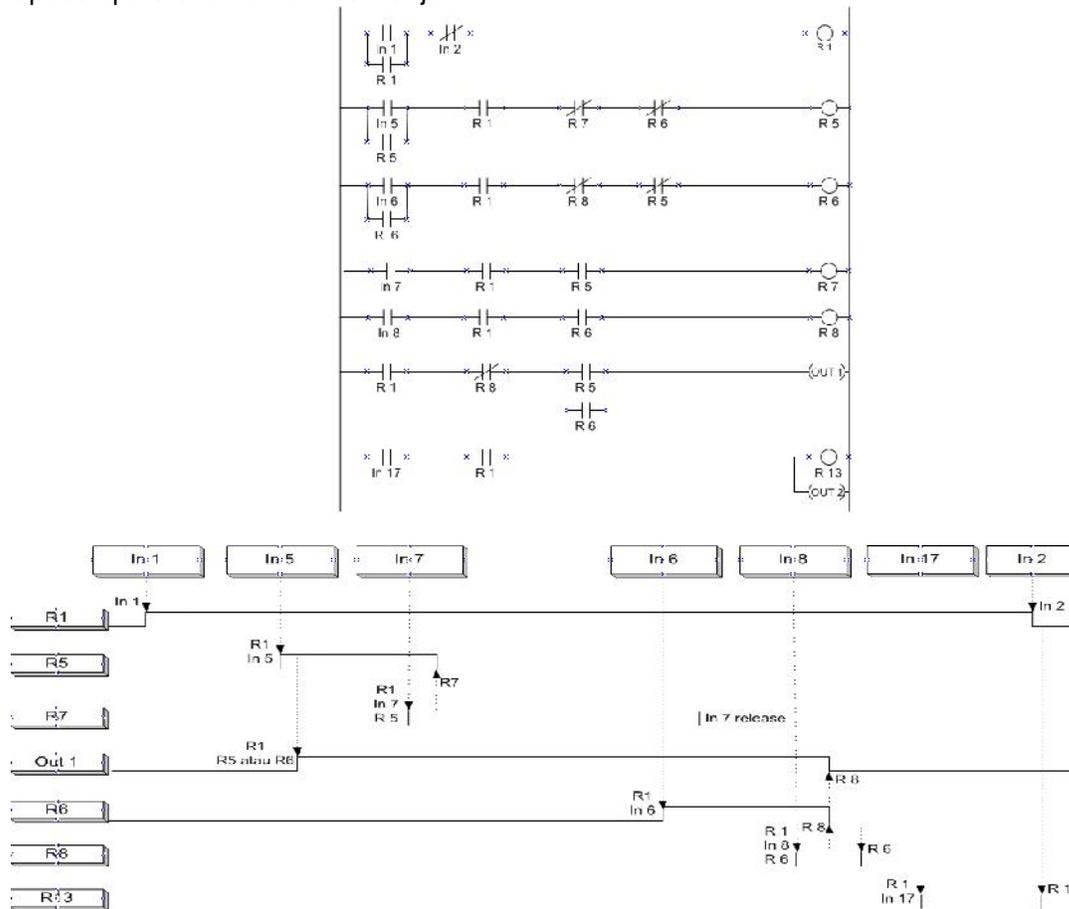


Variabel Input	Nilai variabel input	
In 1 : tombol Start	1: ditekan	0: tidak ditekan
In 2 : tombol Start	1: ditekan	0: tidak ditekan
In 5 : Masukan/ Tekan Tombol Open Source	1: ada masukan/ditekan	0: tidak ada masukan/ tidak ditekan
In 6 : Masukan/ Tekan Tombol Closed Source	1: ada masukan/ ditekan	0: tidak ada masukan/ tidak ditekan
In 7 : Limit Switch Open	1 : tertekan	0 : tidak tertekan
In 8 : Limit Switch Closed	1 : tertekan	0 : tidak tertekan
In 15 : Masukan Indikator Open	1: ada masukan	0 : tidak ada masukan
In 16 : Masukan Indikator Closed	1: ada masukan	0 : tidak ada masukan

Gambar 6. State Diagram dari RLL desain A dan tabel Transisinya

Pada gambar 6 dapat terlihat setelah keadaan (state) "start" (In 1 =1), kumparan relay R1 energized, dengan demikian semua kontak relay R1 bekerja. Dengan bekerjanya semua kontak relay R1, kumparan relay R7 dan R8 (untuk limit switch, In 7=1, In 8 =1) serta kumparan relay R12 dan R13 (indikator, In 15=1 dan In 16 =1) kemungkinan akan bekerja. Pada kondisi pengkabelan kegiatan ini, In 7 merupakan masukan dari open limit switch yang tertekan. In 8 merupakan masukan dari closed limit switch yang tertekan. Sedangkan In 15 merupakan masukan umpan balik yang diperoleh dari keluaran kontak relay R5. Begitu pula dengan In 16, merupakan masukan umpan balik dari keluaran kontak relay R6. Pada praktik percobaan (pengujian) telah didapatkan hasil yang sesuai dengan tahapan pekerjaan. Pada kenyataannya ada kemungkinan perintah masukan input-input tersebut bekerja tidak sesuai dengan harapan. Dalam hal ini ada kemungkinan masukan input terjadi kesalahan dikarenakan kontak input tertekan atau kontak input mengalami kerusakan sehingga memberikan perintah yang tidak diharapkan. Sebagai contoh setelah tombol start ditekan kumparan relay R1 energized. Lampu indikator sumber dapat akan tetap mati atau akan menyala meskipun tombol Source Open belum ditekan (In 5). Disisi lain ada juga kemungkinan motor mati karena kumparan relay R7 atau R8 aktif sehingga kontak relay R7 atau R8 juga aktif. Dengan kata lain kumparan relay R7 atau R8 bekerja hanya tergantung pada masukan In 7 atau In 8. Meskipun secara pengawatan masukan In 7 berasal dari keluaran R5 tetapi error perintah masukan dimungkinkan terjadi. Dilihat dari sisi pemrograman PLC tidak memenuhi perintah secara sekuensial (berurutan). Akibat dari kejadian ini adalah motor akan tetap mati meskipun tombol/masukan open source diaktifkan karena kontak relay R7 bekerja. Peristiwa-peristiwa tersebut kemungkinan akan terjadi ketika kerusakan pada masukan input meskipun secara pengkabelan telah dilakukan dengan benar dan langsung dan operator melakukan kegiatan secara berurutan. Namun secara pemrograman perangkat lunak masih belum sempurna. Selain itu dari sisi pemrograman RLL sudah tidak sekuensial dikarenakan ada kemungkinan proses yang dapat saling mendahului.

Berangkat dari ketidaksempurnaan ini, dibuat sebuah desain pemrograman RLL perbaikan. Desain RLL perbaikan ini diharapkan memenuhi tahapan kegiatan secara berurutan (sekuensial) baik dari sisi pengawatan maupun dari sisi pemrograman komputer. Selain itu, diharapkan adanya suatu perbaikan lebih baik di sisi indikator. Motor penggerak tutup ini menempati bagian penting dalam makalah ini dan motor tersebut dicatu terpisah dengan menggunakan akumulator. Hal-hal yang tidak diinginkan yang berakibat kerusakan mungkin saja terjadi pada motor tersebut. Misalkan ketika sumber terbuka terjadi catu daya motor tidak bekerja optimal. Dibutuhkan informasi mengenai kerusakan ini. Desain yang baru diharapkan dapat memberikan sinyal kerusakan motor kepada operator untuk ditindak lanjuti.



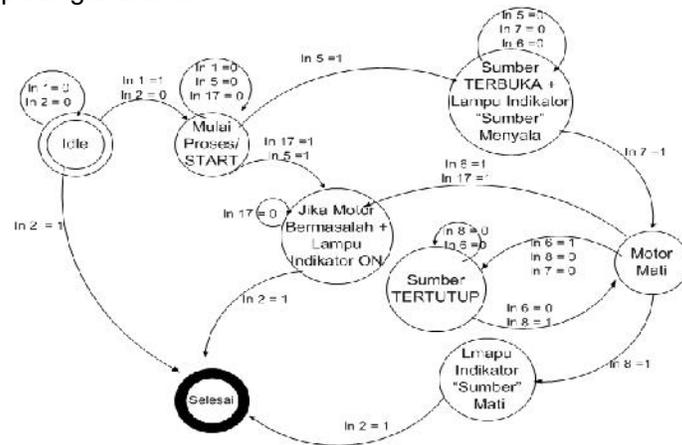
Gambar 8. RLL desain B dan diagram waktunya

Gambar 8 merupakan desain RLL perbaikan. Pada *rung* 4 terdapat penambahan kontak *relay* R5. Kontak *relay* ini memastikan bahwa kumparan *relay* R7 akan bekerja setelah kumparan *relay* R5 bekerja. Ini memberikan arti bahwa setelah diberi perintah masukan *open source* (tombol ditekan) kemudian menunggu *limit switch open* tertekan dan menyebabkan kumparan *relay* R7 bekerja. Dengan bekerjanya kumparan *relay* R7 maka kontak *relay* R7 yang berada pada *rung* 2 akan bekerja sehingga menyebabkan kumparan *relay* R5 akan *de-energized*. Hal ini menyebabkan motor mati. Tentu berbeda dengan RLL desain A..

Indikator sumber terbuka (Out 1) pada desain B hanya digunakan satu buah karena lebih menyangkut aspek keselamatan bukan operasi (indikator bekerja dan indikator tidak bekerja). Desain B, indikator akan menyala bergantung pada tombol *open source* yang menjalankan kumparan *relay* R1. Dan indikator akan mati ketika kumparan *relay* R8 bekerja. Jadi indikator ini tidak tergantung pada tombol *closed source*.

Pada gambar 9. Output R1, R5, dan R7 hampir sama dengan desain A (gambar 7) . Output 1 (pada desain A adalah R12) memiliki hasil yang berbeda dengan desain B. Pada desain ini Out 1 diaktifkan oleh input In 5 dan akan tetap aktif meskipun input In 6 ditekan. Out 1 ini akan tidak aktif ketika kumparan *relay* R8 bekerja. Desain B ini juga menghilangkan lampu indikator operasi *Source Open*. Lampu indikator operasi dijadikan menjadi sebuah lampu peringatan sumber radiasi terbuka. Dimana ketika sumber terbuka lampu ini akan menyala. Lampu ini akan mati ketika sumber benar-benar tertutup dalam hal ini, *limit switch Source Closed* tertekan (R8 aktif =1). *Output* R13 merupakan *output* untuk indikator dan kontak *relay* motor bermasalah. Ketika motor penggerak tutup kolimator mengindikasikan tegangan tidak optimal yang menyebabkan motor tidak bekerja, akan ada sinyal melalui In 17 yang mengaktifkan keluaran R13 ini. Dari kejadian ini akan terlihat lampu indikator motor bermasalah menyala dan akan terkirim sinyal yang mengindikasikan motor bermasalah ke PLC. Dari sini motor sudah pasti dalam keadaan diam tidak bekerja dan membutuhkan bantuan operator datang ke lokasi untuk memperbaiki kerusakan.

Desain B ini diharapkan dapat memperbaiki kekurangan seperti yang sudah dipaparkan dalam penjelasan gambar 6. RLL desain B dituangkan ke dalam diagram *state diagram* pada gambar 9.



Variabel Input	Nilai variabel input	
In 1 : tombol <i>Start</i>	1: ditekan	0: tidak ditekan
In 2 : tombol <i>Stop</i>	1: ditekan	0: tidak ditekan
In 5 :Masukan/ Tekan Tombol <i>Open Source</i>	1: ada masukan/ditekan	0: tidak ada masukan/ tidak ditekan
In 6 : Masukan/ Tekan Tombol <i>Closed Source</i>	1: ada masukan/ditekan	0: tidak ada masukan/tidak ditekan
In 7: <i>Limit Switch Open</i>	1 : tertekan	0 : tidak tertekan
In 8 : <i>Limit Switch Closed</i>	1 : tertekan	0 : tidak tertekan
In 17: Masukan Indikator <i>Open</i>	1: ada masukan	0 : tidak ada masukan

Gambar 9. *State Diagram* dari RLL desain B dan tabel transisinya

Gambar 9 ini merupakan diagram *state diagram* dari RLL desain B. Pada *state diagram* ini terlihat ada beberapa penambahan *state* (keadaan), yaitu *state* jika motor bermasalah+lampu indikator *ON*. Pada desain B ini hanya menggunakan satu indikator sumber yaitu Indikator *Open* (In 17). Hal ini dikarenakan untuk menambah efek kehati-hatian. Indikator keselamatan dalam hal ini adalah In 7 haruslah memberikan perhatian pada lingkungan sekitarnya. Desain B ini tidak menempatkan indikator operasi dimana ada indikator *open* dan indikator *closed*. Jika proses dalam kondisi operasi maka indikator akan menyala dan perlu diperhatikan. Jika proses telah selesai operasi dan keadaan lingkungan sudah aman dari bahaya radiasi maka indikator akan mati. Lampu indikator ini akan menyala ketika tombol *start* ditekan dan akan mati ketika *limit switch closed* tertekan (In 8=1). Perbedaan antara RLL desain A dan RLL desain B adalah diawali dengan perpindahan dari *state start* menuju *state* sumber terbuka. Jika berhasil (motor tidak

bermasalah) akan pindah dengan baik melalui masukan In 5 (In 5=1) tetapi jika ada motor bermasalah maka akan masuk ke *state* motor bermasalah melalui In 5 dan In 17 (In 5 dan In 17 =1). Perbedaan lainnya pada desain B ini, setelah tombol *start* ditekan, meskipun *limit switch open* ditekan secara sengaja atau rusak tidak akan membuat motor mati dikarenakan harus berhasil dahulu sampai di *state* sumber terbuka. Motor akan tetap jalan membuka pintu sampai ada indikasi masukan *limit switch* tertekan (In 7 =1). Setelah *limit switch open* tertekan maka motor akan mati. Tahapan ini juga merupakan perpindahan dari *state* sumber terbuka menuju *state* motor mati.

Ketika tombol *closed source* ditekan (In 6 =1) lampu indikator sumber terbuka tidak langsung mati. Tombol ini bertujuan untuk menutup sumber radioaktif. Seperti telah diketahui sebelumnya, pembukaan dan penutupan menggunakan motor listrik. Motor listrik pada saat ini pun ada kemungkinan bermasalah. Serupa dengan kejadian ketika ditekan tombol *start*, jika ada kemungkinan motor bermasalah maka *state* akan menuju ke keadaan lampu indikator jika motor bermasalah (In 17 =1). Masukan In 17 ini berasal dari sebuah *relay* yang berada pada motor. *Relay* ini, selain akan menghidupkan lampu indikator untuk beberapa saat (dikarenakan catuan motor bermasalah), juga akan memberikan sinyal masukan ke PLC memberikan informasi dan perhatian kepada operator. Operator akan mendatangi lokasi untuk memeriksa dan memperbaiki keadaan di tempat tersebut. Jika tidak ada kondisi motor bermasalah maka motor akan bergerak menutup kolimator sampai menyentuh *limit switch closed*. Ketika *limit switch closed* tertekan (In 8 =1), motor akan mati dan dengan masukan yang sama (In 8=1) lampu *indikator* sumber juga mati. Pada keadaan ini sudah tidak ada lagi sumber radioaktif yang keluar. Lingkungan sudah aman dari sumber radioaktif. Jika proses telah selesai, maka tombol stop dapat ditekan (In 2=1).

4. KESIMPULAN

Telah dilakukan evaluasi terhadap pemrograman RLL untuk buka-tutup kolimator sumber radioaktif pada kegiatan perekrutan portal monitor di PRPN. Dari hasil analisis secara pemrograman masih ada kemungkinan proses yang dapat saling mendahului sehingga proses ini akan berjalan tidak sesuai dengan tahapan kegiatan yang diinginkan. Seharusnya pemrograman RLL ini bersifat sekuensial. Ditemukan pula tidak adanya piranti informasi mengenai kerusakan pada motor padahal motor tersebut ada kemungkinan mengalami gangguan. Dari evaluasi tersebut, dilakukan sebuah desain ulang pemrograman RLL dengan berpedoman kepada keberurutan kegiatan (sekuensial) sesuai dengan tahapan kegiatan. Keberurutan ini tidak hanya secara pengkabelan namun juga secara pemrograman. Desain yang baru juga menambahkan indikator motor bermasalah ketika motor mengalami gangguan. Indikator ini juga akan mengirim sinyal ke operator PLC untuk mengambil tindakan. Perbaikan terakhir adalah mengganti dua buah indikator operasi (indikator *source open* dan *indicator source closed*) menjadi sebuah indikator keselamatan.

5. DAFTAR PUSTAKA

- [1]. YULIAN A,dkk, *Uji Line Scan Camera pada Rancang Bangun Sistem Pencitraan Petikemas dengan Teknik Serapan Sinar Gamma*, Prosiding Pertemuan Ilmiah Rekayasa Perangkat Nuklir – BATAN 2011
- [2]. Anonymous, *Interaksi radiasi pengion dengan materi biologi*, <http://www.infonuklir.com/read/detail/132/interaksi-radiasi-pengion-dengan-materi-biologi>, diunduh 23 Mei 2012
- [3]. KEPUTUSAN KEPALA BADAN PENGAWAS TENAGA NUKLIR nomor : 01/Ka-BAPETEN/V-99, *tentang Ketentuan keselamatan kerja terhadap radiasi*, 1999
- [4]. MADDER ANGELICA, *Timed Automaton Models for Simple Programmable Logic Controllers*, Angelika Mader Hanno Wupper University of Nijmegen Computing Science Institute, Netherlands, <http://eprints.eemcs.utwente.nl/1906/01/euromicro.pdf>, diunduh tanggal 29 januari 2013.

- [5]. BRADLEY ALLEN, *Rockwell Automation. SLC 500 and MicroLogix 1000 Instruction Set*, <http://www.freestudy.co.uk/plc/outcome3%20part%201.pdf>, diunduh tanggal 30 januari 2013.
- [6]. ZHENDONG SU, *Automatic Analysis of Relay Ladder Logic Programs*, EECS Department University of California, Berkeley, <http://www.eecs.berkeley.edu/Pubs/TechRpts/1997/CSD-97-969.pdf>, diunduh tanggal 30 januari 2013.
- [7]. FISCHER A.C, *Cripps Newness Interfacing Companion*, Elsevier Ltd 2002, <http://www.sciencedirect.com/science/book/9780750657204>, diunduh tanggal 30 Januari 2013.
- [8]. AIKEN ALEXANDER, FAHNDRICH MANUEL, ZENDONG SHU, *Detecting Races in Relay Ladder Logic Programs*, , EECS Department University of California, Berkeley, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.7158&rep=rep1&type=pdf>, diunduh tanggal 30 januari 2013.
- [9]. JAE ICK LEE , SUNG WOOK CHUN , SOON JU KANG, *Virtual Prototyping of PLC-Based Embedded System using Object Model of Target and Behavior Model by Converting RLL-to-Statechart Directly*, , Journal of Systems Architecture 48 (2002) 17–35, www.elsevier.com/locate/sysarc diunduh 30 januari 2013.
- [10]. O. ROSSI, *Formal Modeling of Timed Function Blocks for the Automatic Verification of Ladder Diagram Programs*, University Laboratory in Automated Production Research. Ph. Schnoebelen Laboratory for Specification et Verification ENS de Cachan, 61 av. du pr´esident Wilson 94235 Cachan cedex <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.2589&rep=rep1&type=pdf> diunduh tanggal 30 Januari 2013.
- [11]. JEAN-MARIE FARINES1 MAX H. DE QUEIROZ1 MATEUS F. DE SOUZA1 ANA MARIA M. Carpes1 *Fran¸ois Vernadat Modeling and Verification of PLC Programs by using FIACRE Tool Chain*, <http://gforge.enseeiht.fr/docman/view.php/52/4341/B6-UFSC-LAAS.pdf> diunduh tanggal 31 januari 2013.
- [12]. IEC 61131-8 *Programable contrlrolers Guidelines for the application and implementation of programming languages* International Electrotechnical Commission,2003.
- [13]. IEC 61131-3 *Programable contrloler- Programing language*, International Electrotechnical Commission,2003.
- [14]. http://plctext.com/sitebuildercontent/sitebuilderfiles/chapter_4.pdf diunduh tanggal 1 Februari 2013.
- [15]. SUNTORO A., *Pengendalian Pola-Gerak Carrier pada Iradiator Gamma ISG-500*, Prosiding Pertemuan dan Presentasi Ilmiah Penelitian Dasar Ilmu Pengetahuan dan Teknologi Nuklir, PTAPB - BATAN., Yogyakarta, 2010.
- [16]. TAYLOR BOOTH, *Sequential Machines and Automata Theory*, John Wiley and Sons, New York 1967.
- [17]. BELZER JACK, ALBERT GEORGE HOLZMAN, ALLEN KENT,*Encyclopedia of Computer Science and Technology*, Vol. 25. USA: CRC Press. pp. 73. ISBN 0824722752, 1975.
- [18]. KOSHY, THOMAS, *Discrete Mathematics With Applications*. Academic Press. pp. 762. ISBN 0124211801, 2004.
- [19]. WRIGHT, DAVID R., *Finite State Machines*. CSC215 Class Notes. Prof. David R. Wright website, N. Carolina State Univ.. Retrieved July 14, 2012.
- [20]. ATMOKO DIAN F, YULIAN A, MASBATIN I, HANDONO K, *Rancang Bangun Sistem Instrumentasi dan Kendali (SIK) pada Perekayasaan Pencitraan Peti Kemas*, Prosiding Pertemuan Ilmiah Rekayasa Perangkat Nuklir PRPN – BATAN, 12 November 2012.
- [21]. ATMOKO DIAN F, *Perekayasaan Sistem Pencitraan Peti Kemas dengan Teknik Serapan sinar Gamma: Sistem Instrumentasi dan Kendali Jarak Jauh*, Dokumen Perekayasa Teknikal Note TN-08-WP3-RPN-2012-01-440302, PRPN BATAN, 2012.