

IMPLEMENTASI ALGORITMA KONTROL PADA PLC BERBASIS JALA PETRI SINYAL TERINTERPRETASI (JPST)

Hadi Prasetyo

PPPPTK BMTI Bandung
e-mail: prasetiyo@yahoo.com

ABSTRACT

Softwares in plant are important part in system operation and control of hazard as a safety-critical function. Software does not fail in the same manner as hardware. Software failures in control system algorithm are different to hardware failures. Failures in softwares are generally caused by failure in satisfying the required specifications or anticipating unexpected conditions that lead to hazard. Therefore a tool to overcome this problem is required. This papers concentrated in analyzing discrete event control software developed by using Signal Interpreted Petri Net (SIPN) method that used to control an example plan that is "Mixing Tank". This approach was implemented by disaining control software first then evaluate the control software using SIPN control algorithm. A control algorithm is said correct if it fulfil several formal criteria. These formal properties can be derived from SIPN analysis and the results also are given in this paper. Implementation of SIPN based control design on a mixing tank is carried out with computer simulation. Simulation showed that the controlled system satisfies liveness and boundedness properties, and safety conditions. It indicates that there is no deadlock in the places and the transitions. The analysis results showed the controlled mixing tank system has liveness and boundedness properties, and satisfies safety conditions. Transparency value of the developed SIPN is 0.892.

Keywords: Programmable Logic Controller, Signal Interpreted Petri Net, Mixing Tank.

ABSTRAK

Perangkat lunak kontrol dalam sebuah kontrolan (plan) adalah salah satu bagian yang sangat penting, baik dalam hal operasi maupun kontrol terhadap bahaya sebagai fungsi kritis dari keselamatan (*safety*). Kegagalan perangkat lunak kontrol mempunyai karakteristik yang tidak sama dengan perangkat keras. Kegagalan perangkat lunak kontrol umumnya merupakan kesalahan dalam menentukan persyaratan yang ditetapkan, atau gagal dalam mengantisipasi kondisi yang mengakibatkan bahaya. Oleh karena itu diperlukan sebuah pendekatan yang dapat digunakan untuk mengatasi masalah ini. Dalam paper ini akan dipaparkan analisis perangkat lunak kejadian diskrit dengan menggunakan Jala Petri Sinyal Terinterpretasi (JPST) sebagai sebuah pendekatan pemrograman PLC yang akan digunakan untuk mengontrol sebuah contoh plan "Tangki Pengaduk". Pendekatan ini diimplementasikan dengan cara terlebih dahulu merancang perangkat lunak kontrol, kemudian mengevaluasi perangkat lunak kontrol tersebut dengan algoritma kontrol JPST. Sebuah algoritma kontrol dikatakan benar apabila telah memenuhi beberapa kriteria formal. Sifat-sifat formal ini dapat diperoleh melalui analisis dengan menggunakan pendekatan JPST yang hasilnya akan diberikan pada paper ini. Pada paper ini juga dipresentasikan implementasi rancangan kontrol pemodelan JPST sistem Tangki Pengaduk dalam bentuk simulasi komputer menggunakan diagram Ladder. Pada simulasi menunjukkan bahwa sistem kontrol memenuhi sifat-sifat: daya hidup dan kondisi aman (*safety*) karena tidak ditemukan kondisi terkunci pada transisi dan *place*. Nilai transparansi yang diperoleh adalah 0,892.

Kata Kunci: Pemrograman, Jala Petri Sinyal Terinterpretasi, Tangki Pengaduk.

1. PENDAHULUAN

Jala Petri telah diperkenalkan untuk pertama kalinya oleh Dr. Carl Adam Petri pada tahun 1962. Jala Petri ini adalah sebuah alat yang digunakan untuk pemodelan dan analisis formal dalam bidang komputer, rekayasa sistem dan disiplin ilmu lainnya. Jala Petri mengkombinasikan sebuah definisi teori matematika dengan sebuah representasi grafik tingkah laku dinamik sistem [1]. Dari sisi aspek teori Jala Petri menjadikan pemodelan dan analisis sistem menjadi lebih teliti, sementara representasi grafik Jala Petri dapat memvisualisasikan perubahan keadaan dari sistem yang dimodelkan. Jala Petri adalah salah satu alat yang digunakan dalam pemodelan algoritma kontrol. Dengan menggunakan Jala Petri ini dapat membantu dalam penganalisisan kontrol secara keseluruhan [2]. Pemrograman yang didisain dengan menggunakan metoda Jala Petri akan menghasilkan disain program yang terstruktur dan terorganisasikan secara baik, sehingga akan lebih mudah dalam melakukan evaluasi, perubahan, pemeliharaan dan sebagainya. Jala Petri dapat juga dilengkapi dengan alat bantu berupa grafik yang berfungsi sebagai media komunikasi antara pengguna, dalam hal ini perancang dan pengguna.

Sebuah Jala Petri secara formal didefinisikan sebagai sebuah *tuple-5*, yaitu:

$$N = (P, T, I, O, M_0) \tag{1}$$

dimana:

$P = \{p_1, \dots, p_n\}$, himpunan dari *place*,

$T = \{t_1, \dots, t_m\}$, himpunan dari transisi,

$I : (p, t)$ adalah sebuah pemetaan

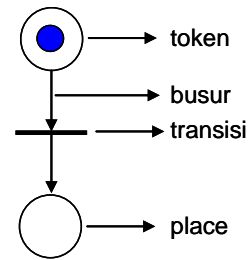
$P \times T \rightarrow \{0,1\}$, yaitu himpunan arah busur dari *place* ke transisi.

$O : (t, p)$ adalah sebuah pemetaan

$T \times P \rightarrow \{0,1\}$, yaitu himpunan arah busur dari transisi ke *place*.

$M_0 : P \rightarrow N$, adalah merupakan penandaan awal.

Jala Petri pada dasarnya terdiri dari tiga obyek, yaitu: *place*, transisi, dan arah busur. Secara grafik Jala Petri dapat direpresentasikan, sebagai berikut:



Gambar 1: Jala Petri Sederhana

Oleh karena pada model Jala Petri pada umumnya tidak terdapat hubungan antara algoritma dan lingkungan, maka untuk memecahkan masalah ini digunakan model JPST yang merupakan perluasan dari model Jala Petri. Pada model JPST ini transisi dihubungkan dengan kondisi pemicuan yang tergantung pada kejadian eksternal.

Transisi dalam JPST diasosiasikan dengan kondisi pemicuan seperti yang diberikan pada sinyal masukan fungsi Boolean. *Place* dari sebuah JPST diasosiasikan dengan aksi penetapan sinyal keluaran. JPST digambarkan oleh tuple 9-JPST [3].

$$JPST = (P, T, F, m_o, I, O, \varphi, \omega, \Omega) \tag{2}$$

dimana:

(P, T, F, m_o) seperti Jala Petri pada umumnya dengan tempat (*place*) P , transisi T , busur F , dan penandaan awal biner m_o , dengan $|P|, |T|, |F| > 0$.

I : sebuah himpunan dari logika sinyal masukan dengan $|I| > 0$.

O : sebuah himpunan dari logika sinyal keluaran dengan $I \cap O = \emptyset, |O| > 0$.

φ : sebuah asosiasi pemetaan dari setiap transisi $t_i \in T$ dengan sebuah kondisi pemicuan $\varphi(t_i) =$ fungsi Boolean (I)

ω : sebuah asosiasi pemetaan dari setiap *place* $p_i \in P$ dengan sebuah keluaran $\omega(p_i) \in (0,1,-)^{|O|}$, dimana tanda (-) berarti tidak dipedulikan.

Ω : fungsi keluaran yang menggabungkan keluaran ω dari semua penanda *place* $\Omega : m \rightarrow (-,1,0,c)^{|O|}$. Kombinasi keluaran dapat dalam keadaan tidak didefinisikan (-), satu (1), kosong (0), atau kontradiksi (c)

Aliran token menunjukkan tingkah laku dinamik dari sebuah JPST yang bergerak melalui jala-jala, ini dapat dilihat dari perubahan tandanya. Aliran ini direalisasikan

oleh pemicuan transisi. Pemicuan sebuah transisi t_i melepas sebuah token dari masing-masing tempat (*place*) sebelumnya $(p_j, t_i) \in F$ dan meletakkan sebuah token pada masing-masing tempat setelahnya $(t_i, p_j) \in F$. Setelah pemicuan transisi, sinyal keluaran dihitung kembali dengan menerapkan Ω ke penanda. Untuk proses pemicuan terdapat empat ketentuan:

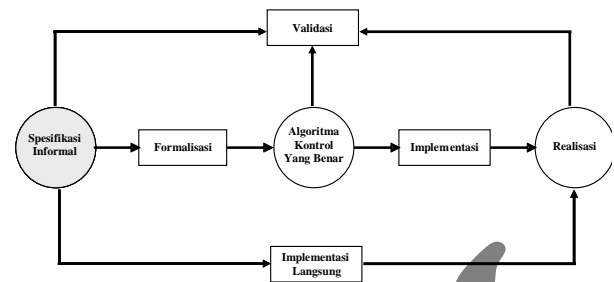
- 1) Transisi dapat dilakukan, jika semua tempat asal sebelum pemicuan (*pre-place*) ditandai dan semua tempat yang dituju (*post place*) tidak bertanda.
- 2) Pemicuan transisi dapat dilakukan segera, ketika kondisi pemicunya dipenuhi $o(t_i) = \text{true}$ (benar).
- 3) Semua transisi yang dapat memicu dan tidak konflik dengan transisi yang lain secara simultan.
- 4) Proses pemicuan diiterasi sampai sebuah penanda stabil tercapai (yaitu sampai tidak ada transisi dapat yang memicu lagi). Jika pemicuan transisi diperkirakan tidak mengambil waktu, iterasi pemicuan diinterpretasikan seperti bersamaan.

Ketentuan pemicuan ini berbeda dengan apa yang ada di Jala Petri standar, oleh karenanya algoritma khusus digunakan untuk kalkulasi yang didasarkan pada gambar ketercapaian JPST.

2. METODE

Tujuan utama penerapan metoda formal pada disain program PLC adalah untuk mendapatkan sebuah algoritma kontrol yang benar sebelum pengimplementasian (lihat gambar 2). Jika tujuan ini tercapai, realisasi dari algoritma pada PLC harus tidak boleh terdapat kesalahan formal atau fungsional. Hanya masalah yang tersisa yang boleh muncul pada saat atau selama instalasi, sebagai contoh seperti masalah pewaktu (*timing*) dalam perangkat keras PLC. Semua kesalahan yang lain harus sudah ditemukan dalam tahap awal selama proses disain formal. Ini penting sebab semakin lebih awal masalah ditemukan, semakin lebih mudah,

lebih cepat, dan lebih murah untuk dipecahkan.

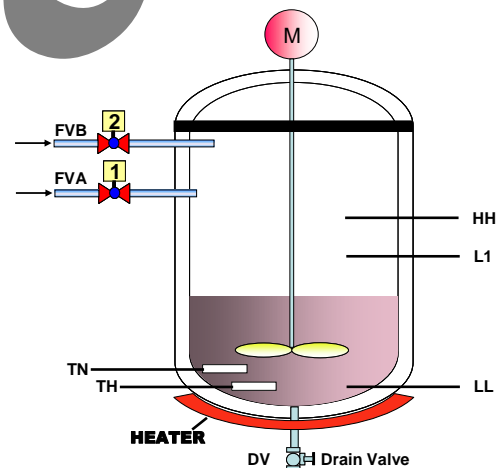


Gambar 2: Proses Disain

3. FORMALISASI SPESIFIKASI INFORMAL

3.1 Sifat-sifat Masalah Fungsional Spesifik

Untuk menghasilkan tingkah laku mesin pengaduk bekerja secara otomatis, algoritma kontrol didisain menggunakan spesifikasi informal sebagai berikut:



Gambar 3: Tangki Pengaduk

- 1) Jika tombol start ditekan (lihat Tabel 1) dan LL tercapai, katup pengisian (FVA) terbuka sampai level L1 tercapai dalam waktu bersamaan motor pengaduk bekerja (motor pengaduk akan terus bekerja dan akan berhenti jika tahap proses pengosongan tercapai).
- 2) Jika L1 tercapai, katup pengisian (FVB) terbuka sampai level HH tercapai.
- 3) Jika HH tercapai, proses dilanjutkan dengan pemanasan fluida (proses pemanasan tidak dilakukan jika temperatur fluida sama dengan mengatur temperatur).

- 4) Takaran bahan-bahan yang dicampur harus sesuai dengan yang telah ditetapkan.
- 5) Setelah proses pemanasan selesai katup pengosongan (DV) terbuka.
- 6) Jika pengosongan selesai, skenario berulang seperti semula secara otomatis, dan proses akan berhenti jika tombol tekan stop ditekan.
- 7) Jika dalam proses pemanasan temperatur yang dicapai melebihi nilai pengaturannya, maka sistem akan memberikan sinyal tanda bahaya (alarm). Dalam keadaan seperti ini proses akan berhenti secara otomatis setelah proses pengosongan tangki selesai.
- 8) Dalam keadaan terdapat kesalahan selama proses pemanasan bahan adukan sehingga menghasilkan temperatur yang berlebihan, maka jika sistem tidak dimatikan akan dapat menimbulkan bahaya kecelakaan. Kontrolan ini diasumsikan mempunyai potensi bahaya yang dapat menimbulkan kecelakaan dan dapat mengakibatkan resiko kematian personel.

3.2 Formalisasi Algoritma Kontrol

Seperti telah disebutkan diatas bahwa terdapat beberapa langkah yang dilibatkan dalam pengembangan sebuah program kontrol yang benar. Untuk melakukan formalisasi algoritma kontrol digunakan jenis khusus Jala Petri, yaitu: JPST [4]. JPST adalah sebuah Jala Petri dengan penanda biner dan perluasan (extension):

- Setiap transisi digabungkan dengan sinyal masukan fungsi Boolean (kondisi pemicuan).
- Setiap *place* digabungkan dengan fungsi keluaran, yang menetapkan sub-set dari sinyal keluaran yang tempatnya ditandai.

Tingkah laku dinamik JPST diindikasikan oleh penanda atau *token* yang mengalir melalui jala-jala. Aliran *token* ini direalisasikan oleh pemicuan transisi, pemicuan transisi melepas penanda dari masing-masing tempat sebelumnya dan

menempatkan penanda pada masing-masing tempat berikutnya.

Untuk proses pemicuan terdapat empat ketentuan:

- 1) Sebuah transisi diperkenankan jika semua tempat asal ditandai dan semua tempat tujuan tidak ditandai.
- 2) Sebuah transisi memicu dengan segera jika dapat dipicu dan kondisi pemicuannya dipenuhi.
- 3) Semua transisi yang dapat dipicu dan tidak dalam konflik dengan pemicuan transisi yang lain secara serentak.
- 4) Proses pemicuan diiterasi sampai sebuah penanda stabil tercapai (sampai tidak ada transisi dapat memicu lagi tanpa sebuah perubahan dari nilai sinyal masukan). Iterasi pemicuan diinterpretasikan secara serentak. Ini berarti bahwa sebuah perubahan dari nilai sinyal masukan tidak dapat terjadi selama proses pemicuan.

Setelah penanda stabil tercapai, sinyal keluaran dihitung dengan mengevaluasi fungsi keluaran dari *place* yang ditandai.

3.3 Masukan dan Keluaran Sistem Kontrol Kejadian Diskrit Tangki Pengaduk

Didasarkan pada prosedur operasi tangki pengaduk, maka dapat dibuat definisi formal, sebagai berikut:

Masukan yang diperoleh dari sistem kontrol berasal dari tombol-tombol operasi dan sensor-sensor yang ada di tangki pengaduk. Sedangkan keluaran yang diinginkan berupa hasil aksi pemicuan dan penonaktifan pada *place*. Berikut adalah daftar masukan dan keluaran yang didefinisikan dalam sistem kontrol tangki pengaduk.

Tabel 1 Daftar Masukan

Nomor Masukan	Nama	Definisi	Syarat
1	Start	Tombol menghidupkan tangki pengaduk	Ditekan
2	Stop	Tombol mematikan tangki pengaduk	Ditekan
3	LL	LowLow water level	Level < LL
4	L1	Low water level fluida 1	Level = L1
5	HH	High water level fluida 1 dan 2	Level = HH
6	TN	Pengaturan temperatur	Temperatur = TN
7	TH	Temperatur tinggi (high)	Temperatur ≥ TH

Tabel 2 Daftar Keluaran

Nomor Keluaran	Nama	Definisi
1	FVA	Katup pengisian fluida 1
2	FVB	Katup pengisian fluida 2
3	MM	Motor pengaduk
4	DV	Katup pengosongan
5	HTR	Pemanas
6	ALR	Tanda bahaya (alarm)

3.4 Pemodelan JPST

Pemodelan dapat dilakukan setelah mendapatkan informasi dari cara kerja operasi dari tanki pengaduk. Untuk melakukan

pemodelan dengan menggunakan JPST terlebih dahulu harus mendefinisikan *place* dan transisi [1]. Berikut adalah definisi dari *place* dan transisi tersebut.

Tabel 3 Daftar Place

Pi	Definisi	$\omega(Pi)$
1	Tangki pengaduk siap dioperasikan (<i>standby</i>)	(1,0,0,0,0,0,0,0,0)
2	Aux 1	(0,1,0,0,0,0,0,0,0)
3	Pengisian fluida 1	(0,0,1,0,0,0,0,0,0)
4	Pengadukan fluida 1 dan 2	(0,0,0,1,0,0,0,0,0)
5	Pengisian fluida 2	(0,0,0,0,1,0,0,0,0)
6	Pemanasan	(0,0,0,0,0,1,0,0,0)
7	Pengosongan tangki	(0,0,0,0,0,0,1,0,0)
8	Aux 2	(0,0,0,0,0,0,0,1,0)
9	Pengosongan tangki & tanda bahaya (alarm)	(0,0,0,0,0,0,0,0,1)

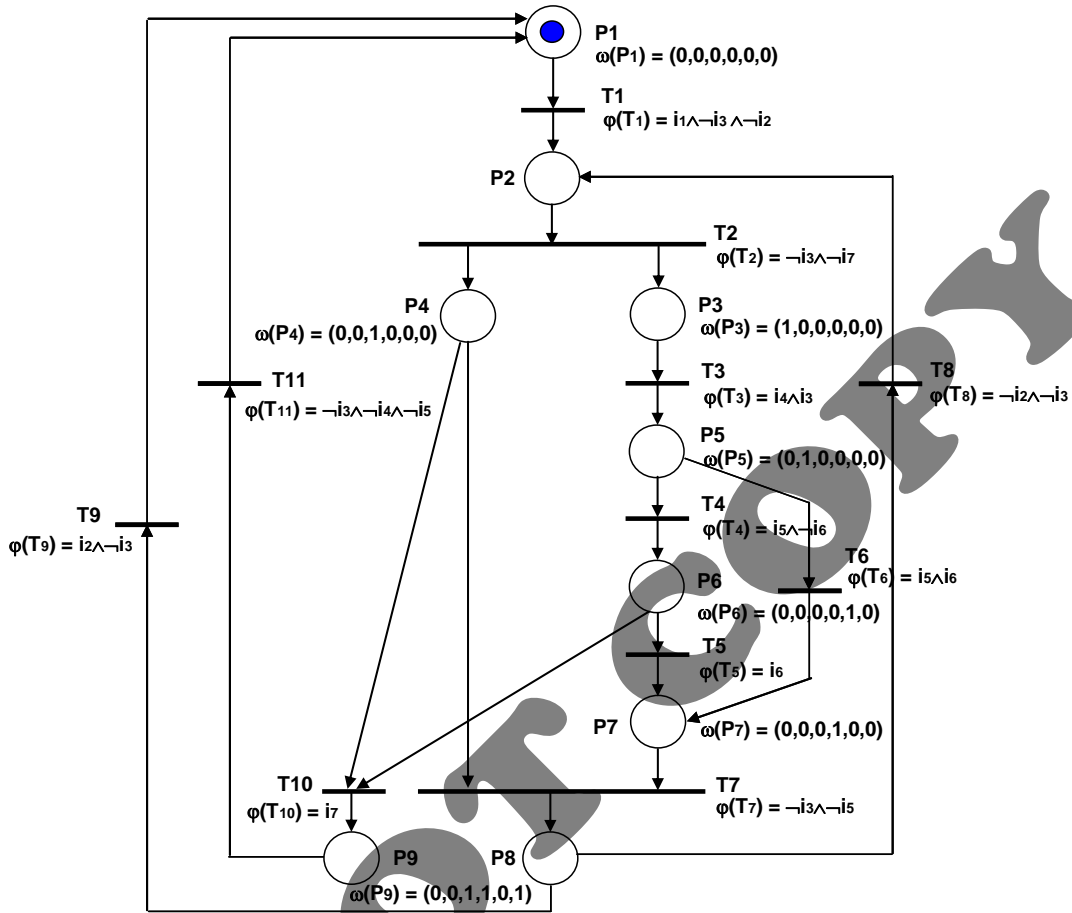
Tabel 4 Daftar Transisi

Transisi	Kondisi Prasyarat Masukan	Definisi	Place Asal	Place Tujuan
t1	$i_1 \wedge \neg i_3 \wedge \neg i_2$	Penekanan tombol start	P1	P2
t2	$\neg i_3 \wedge \neg i_7$	Tangki kosong	P2	P3 dan P4
t3	$i_4 \wedge i_3$	Level fluida 1 cukup	P3	P5
t4	$i_5 \wedge \neg i_6$	Level fluida 1 dan 2 cukup	P5	P6
t5	i_6	Pemanasan selesai	P6	P7
t6	$i_5 \wedge i_6$	Temperatur sesuai pengaturan (setting)	P5	P7
t7	$\neg i_3 \wedge \neg i_5$	Proses pengosongan dan pengadukan tangki selesai	P4 dan P7	P8
t8	$\neg i_2 \wedge \neg i_3$	Proses diulangi (otomatis)	P8	P2
t9	$i_2 \wedge \neg i_3$	Proses dihentikan	P8	P1
t10	i_7	Temperatur melebihi pengaturan	P4 dan P6	P9
t11	$\neg i_3 \wedge \neg i_4 \wedge \neg i_5$	Proses terhenti	P9	P1

3.5 Model Perancangan Tangki Pengaduk

Aksi keluaran pada *place* ω (π) berupa kode bilangan 1 dan 0. Bilangan 1

menunjukkan keluaran dalam keadaan aktif *on* dan bilangan 0 menandakan keluaran dalam keadaan tidak aktif *off*.



Gambar 4: Model Rancangan Tangki pengaduk

4. ANALISIS JPST

Secara umum terdapat empat sifat yang dapat diperiksa dengan menggunakan analisis JPST, keempat sifat tersebut terkait dengan grafik jala ketercapaian. Adapun keempat sifat dalam definisi JPT dan relevansinya dengan pengontrol, adalah: ketercapaian (*reachability*), daya hidup (*liveness*), dapat kembali ke penandaan awal (*reversibility*) dan keterbatasan (*boundedness*) [5].

Ketercapaian berarti bahwa sebuah penanda selalu dapat mencapai lagi ke penandaan awal [2], sebuah ketercapaian dapat segera diperoleh dari penandaan asal ke penandaan tujuan dengan menggunakan persamaan:

$$M'(p) = M(p) + O(t,p) - I(p,t)$$

dengan:

I: ($P \times T$) merupakan sebuah fungsi masukan dari *place* ke transisi.

O: ($T \times P$) merupakan sebuah fungsi keluaran dari transisi ke *place*.

4.1 Analisis Ketercapaian

Dalam JPST sebuah *place* dikatakan aktif apabila dalam *place* tersebut berisi penanda atau token. Pada pemodelan JPST sistem tangki pengaduk ini *place* 1 adalah merupakan *place* yang diberi penanda untuk pertama kalinya (awal), selanjutnya ini akan menghasilkan pemetaan penandaan dari *place* $m_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$. Dari penandaan awal ini akan didapatkan penandaan baru, apabila kondisi pemicuan transisi dipenuhi. Penandaan baru ini dapat diperoleh dengan melakukan perhitungan menggunakan

persamaan matrik ketercapaian, persamaan tersebut adalah :

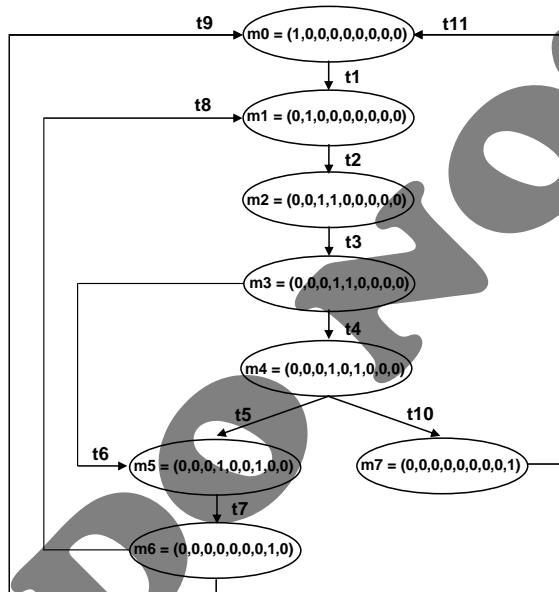
$$M'(p) = M(p) + O(t,p) - I(p,t) \quad (3)$$

Jika semua penandaan dijadikan satu himpunan ketercapaian, adalah sebagai berikut:

$$M'(p_i) = M(p_i) + (OUT * N) - (IN * N), \quad \forall p_i \in \quad (4)$$

$$RS = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]^T, \\ [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]^T, \\ [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^T, \\ [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T.$$

Untuk mendapatkan apakah sistem yang dimodelkan dapat mencapai keadaan yang spesifik, maka diperlukan/dicari sebuah urutan pemicuan transisi dari penandaan awal m_0 ke penandaan baru m_i . Perubahan penandaan dari *place* asal ke *place* baru yang terjadi pada pemodelan system JPST dapat dilihat dari grafik ketercapaian seperti terlihat pada Gambar 5.



Gambar 5: Grafik Ketercapaian

Dari gambar grafik ketercapaian terlihat bahwa suatu penandaan baru dapat tercapai jika dan hanya jika dari penandaan sebelumnya memenuhi syarat kondisi terpicunya transisi. Sebagai contoh penandaan m_0 akan berpindah ke penandaan m_1 jika syarat pemicuan transisi t_1 terpenuhi.

4.2 Analisis Keterbatasan

Untuk memeriksa sifat keterbatasan dapat dilihat dari hasil perhitungan P-invarian dari sistem [6], dimana JPST yang disertai dengan positif P-invarian dikatakan memiliki sifat keterbatasan. Sedangkan untuk memenuhi sifat keselamatan, maka jumlah token dalam *place* tersebut tidak melebihi bilangan integer 1 untuk setiap penandaan M yang dicapai [7,12].

Dari hasil perhitungan P-invarian pemodelan sistem ini ternyata dihasilkan nilai [110100011] dan [111011111], dan persamaan token-nya dapat dituliskan sebagai berikut:

$$M(p_1) + M(p_2) + M(p_4) + M(p_8) + M(p_9) = 1$$

$$M(p_1) + M(p_2) + M(p_3) + M(p_5) + M(p_6) + M(p_7) + M(p_8) + M(p_9) = 1$$

Dengan demikian ini membuktikan bahwa jumlah token dalam setiap *place* adalah 0 dan 1, sehingga disini dapat dikatakan bahwa pemodelan sistem yang telah dirancang mempunyai sifat keterbatasan dan keselamatan.

4.3 Analisis Daya Hidup

Sifat daya hidup berkaitan erat dengan situasi terkunci. Sebuah JPST dikatakan hidup jika dan hanya jika transisinya hidup, atau dengan kata lain bahwa sistem memiliki daya hidup jika semua penandaan yang dicapai dari keadaan awal sekurang-kurangnya memiliki satu penandaan yang bisa dituju dan menjadi penandaan yang dituju dari penandaan lainnya. Oleh karena itu dalam pemodelan JPST harus menghasilkan situasi yang tidak boleh terkunci.

Untuk memeriksa sifat daya hidup ini dapat dilihat dari hasil perhitungan T-invarian [6], jika hasil perhitungan T-invarian adalah positif maka sistem dikatakan mempunyai sifat daya hidup dan sifat keterbatasan. Dari hasil perhitungan T-invarian pemodelan sistem yang telah dirancang ternyata diperoleh 5 T-invarian yang bebas linier, yaitu: $\sigma_1 = [01100111000]$, $\sigma_2 = [11100110100]$, $\sigma_3 = [11110000011]$, $\sigma_4 = [01111011000]$ dan $\sigma_5 = [11111010100]$. Dengan demikian ini mengindikasikan bahwa

pada sistem tidak terdapat transisi yang deadlock, sehingga pemodelan sistem dapat dikatakan memiliki sifat *liveness* dan *boundedness*.

4.4 Analisis Transparansi

Analisis transparansi pada kriteria t_1 , t_6 dan t_7 dapat secara langsung dievaluasi dengan menggunakan JPST. Sedangkan untuk menentukan kriteria t_2 , t_3 , t_4 , t_5 dan t_8 dapat dilakukan dengan menggunakan grafik ketercapaian dan deskripsi yang ada di JPST [8].

Sehingga dari hasil perhitungan dari setiap kriteria transparansi diatas diperoleh nilai rata-rata seperti ditunjukkan pada tabel 5.

Tabel 5 Nilai Transparansi

Kriteria	Ukuran
t_1 komentar	$t_1 = 0 * 0,075 = 0$
t_2 tidak ada masukan trivial	$t_2 = 1 * 0,075 = 0,075$
t_3 tidak ada keluaran trivial	$t_3 = 1 * 0,15 = 0,15$
t_4 tidak ada keluaran redundan	$t_4 = 1 * 0,15 = 0,15$
t_5 keselamatan	$t_5 = 1 * 0,25 = 0,25$
t_6 arah	$t_6 = 19/25 * 0,075 = 0,057$
t_7 interseksi	$t_7 = 4/5 * 0,075 = 0,06$
t_8 tidak ada sinkronisasi	$t_8 = 1 * 0,15 = 0,15$
Nilai transparansi = 0,892	

5. IMPLEMENTASI DAN SIMULASI PEMODELAN JPST

5.1 Implementasi Pemodelan

Untuk melakukan pengubahan JPST ke program diagram ladder digunakan metoda lintasan token logika ladder atau TPLL (*Token Passing Ladder Logic*) [9][10][11]. Tingkah laku dinamik JPST berupa aliran token yang mengalir melalui jala-jala dan

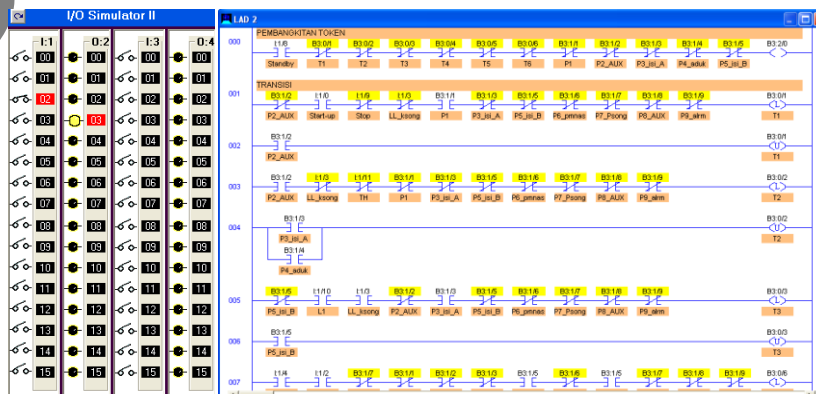
kemudian mengubah penandaannya. Aliran ini direalisasikan dengan pemicuan transisi. Pemicuan transisi akan melepas token dari setiap *place* asal dan memindahkannya ke *place* tujuan. Setelah penandaan yang stabil pada *place* tercapai, maka *place* tersebut dikatakan aktif. Jika *place* ditandai, maka fungsi keluaran yang bersesuaian dengan *place* tersebut dieksekusi. Pembuatan program ladder yang didasarkan pada JPST tersebut dikelompokkan dalam empat bagian, yaitu:

- Pembangkitan token.
- Set dan reset transisi.
- Set dan reset *place*.
- Set dan reset keluaran yang bersesuaian dengan *place*.

5.2 Simulasi Pemodelan

Untuk mengakomodasi pemodelan JPST dari sistem yang ditransformasikan ke dalam diagram ladder yang dikelompokkan kedalam empat kelompok rung yang masing-masing adalah: kelompok rung untuk pembangkitan token (1 rung), kelompok rung untuk transisi (22 rung), kelompok rung untuk *place* (17 rung), dan kelompok rung untuk keluaran (8 rung), lihat gambar 6.

Dari hasil simulasi menunjukkan bahwa pada sistem telah memenuhi syarat sifat keterbatasan dan daya hidup. Ini ditandai dengan tidak adanya *place* dan transisi mengalami kebuntuan. Pada saat program simulasi dijalankan perubahan status dari: transisi, *place* dan keluaran terlihat dari perubahan yang ada pada setiap instruksi keluaran ladder yang dialokasikan untuk transisi, *place* dan keluaran (eksternal).



Gambar 6: Simulasi Pemodelan JPST yang Ditransformasikan ke LLD

6. SIMPULAN

Dalam tesis ini telah dikembangkan sebuah metoda yang digunakan menilai program kontrol PLC yang didasarkan pada hasil analisis sifat-sifat perangkat lunak kontrol dengan menggunakan metoda JPST.

1. Dari hasil perhitungan P-invarian pemodelan sistem ini ternyata dihasilkan nilai [110100011] dan [111011111]. Dengan demikian ini membuktikan bahwa jumlah token dalam *place* pada setiap sub-net adalah 1, sehingga disini dapat dikatakan bahwa pemodelan sistem yang telah dirancang mempunyai sifat keterbatasan dan keselamatan.
2. Diperoleh lima (5) T-invarian yang bebas linier pada pemodelan sistem yang telah dirancang, yaitu:
 $\sigma_1=[01100111000]$, $\sigma_2=[11100110100]$,
 $\sigma_3=[11110000011]$, $\sigma_4=[01111011000]$
dan $\sigma_5=[11111010100]$. Dengan demikian ini mengindikasikan bahwa pada sistem tidak terdapat transisi yang terkunci (*deadlock*), sehingga pemodelan sistem dapat dikatakan memiliki sifat *liveness* dan *boundedness*.
3. Dari hasil simulasi diagram *ladder* yang merupakan transformasi dari JPST menunjukkan, bahwa pada sistem telah memenuhi syarat sifat keterbatasan dan daya hidup. Ini ditandai dengan tidak adanya *place* dan transisi yang mengalami kebuntuan. Pada saat program simulasi dijalankan perubahan status dari: transisi, *place* dan keluaran terlihat dari perubahan (warna) yang ada pada setiap instruksi keluaran *ladder* yang dialokasikan untuk transisi, *place* dan keluaran (eksternal) tersebut.

REFERENSI

[1] Frey, G. (2000). *Analysis of Petri Net Based Control Algorithms Basic Properties*. Proceedings of American

- Control Conference ACC 2000. Chicago, U.S, 3172 – 3176.
- [2] Frey, G. (2000). *Formal methods in PLC programming*. Proceedings of the IEEE Conference on Systems Man and Cybernetic SMC 2000. Nashville, U.S, 2431 – 2432.
- [3] Frey, G, Schettler, HG. (1998). *Algebraic Analysis of Petri Net Based Control Algorithm*. Proceeding of the 4th IEE Workshop on Discrete Evet System WODES'98. Cagliari, Italy.
- [4] Mertke, T, Frey, G. (2001). *Formal Verification of PLC-Program Generated From Signal Interpreted Petri Nets*. Proceedings of the IEEE Systems Man and Cybernetic Conference. Nashville, U.S, 2431 – 2432.
- [5] Starke, P.H., Roch, S. (2002). *Analysing Signal-Net Systems*, Informatik-Bericht 162, Berlin, 89 - 95.
- [6] Wang, J. (1998). *Timed Petri Nets*. Massachusetts: Kluwer Academic Publisher
- [7] Cheol Cho, Y., Hyun Kwon, W. (2000). *Inhibitor Arc ased State Avoidance Controller for Non-convex Forbidden State Problem in Petri Nets*, ERC-ACI and School of Electrical Engineering Soul National University, Soul, Korea.
- [8] Frey, G, Litz, L. (2000). *Transparency Anlysis of Petri Net Based Logic Controller A Measure Software Quality in Automation*. Proceedings of American Control Conference ACC 2000. Chicago, U.S, 3183 – 3186.
- [9] Filer, R., Leinonen, G. (1992). *Programmable Controllers and Designing Sequential Logic*. Florida: Saunders College Publishing
- [10] Sammarco, J. J. (2005). *Programmable Electronic Mining Systems: Best Practice Recommendations*. Cincinnati: NIOSH-Publications.

- [11] Uzam, M., Jones, A.H. (2002). *A New Petri-Net-Based Synthesis Technique for Supervisory Control of Discrete Event Systems*, *Turk J Elec Engin* VOL.10 NO.1 2002, Universitesi, M'uhendislik-Mimarlık Fak'ultesi, Turkey, 20 – 24.
- [12] Tiruchinapalli, K. (2005). *Modular Safety Design of Autonomous Systems Using KURT3D as an Example*, Master Thesis, University of Applied Sciences Bonn-Rhein-Sieg, 32 – 42.

DO NOT COPY