

PEMODELAN *HYBRID* SINTESIS PADA *AUTOMATED MANUFACTURING SYSTEM* (AMS) DENGAN MENGGUNAKAN PETRI NET

DORTEUS LODEWYIK RAHAKBAUW

Staf Jurusan Matematika FMIPA UNPATTI

Jl. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon

e-mail: lodewyik@gmail.com

ABSTRAK

Dalam Jurnal ini dibahas masalah sistem even diskrit yang ada pada *Automated Manufacturing System* (AMS), khususnya menyangkut model, kedinamikan, dan mendesain sistem AMS dengan menggunakan *tool* matematik Petri net. Selanjutnya dikonstruksikan suatu model AMS yang komponen utamanya terdiri dari dua *entry*, dua *exit*, lima mesin, dua *buffer*, empat robot, dan dua sistem *Automated Guided Vehicle* (AGV) yang ada dalam AMS, dimana masing-masing komponen tersebut memiliki deskripsi kerja berlainan sesuai tugas masing-masing. Dengan menggunakan metode *hybrid*, AMS akan ditransformasikan ke dalam *tool* matematik Petri net dengan menggunakan deskripsi sintesis yang didalamnya terdiri atas dua prosedur kerja yaitu *top-down design* dan *bottom-up design*. Dekomposisi (*decomposition*) dan penghalusan (*refinement*) yang meliputi penambahan *place shared* dan *non-shared resource* ke dalam Petri net dibutuhkan untuk mendapatkan desain Petri net AMS yang akan mengurangi atau menghilangkan *deadlock* pada sistem AMS. Dengan demikian, suatu definisi dari *hybrid* sintesis serta struktur Petri net dapat terjawab. Perilaku mendesain sistem ini akan dibahas untuk mendapatkan suatu desain yang efektif dalam fungsi dan penggunaan AMS tersebut.

Keywords: AMS, *Hybrid* sintesis, Petri net, *Coverability tree*.

PENDAHULUAN

Dalam perkembangan Petri net (Murata, 1989) adalah suatu pemodelan formal yang diperkenalkan oleh Carl Adam Petri pada tahun 1962 untuk memodelkan sistem terdistribusi. Petri net cepat berkembang sebagai suatu hal yang menjanjikan dalam pemodelan, sehubungan dengan fungsinya untuk mewakili sejumlah fitur dari perilaku sistem dinamik *event* diskrit. Sistem *event-driven* meliputi manufaktur otomatis, jaringan komunikasi, sistem operasi komputer, sistem informasi kantor, dan lainnya. Mungkin tidak bersamaan dan bertahap namun memperlihatkan beberapa karakteristik, diantaranya: *concurrency*, *conflict*, *mutual exclusion*, dan *non-determinism*. Karakteristik-karakteristik ini cukup sulit jika dideskripsikan menggunakan teori kontrol tradisional yang merujuk pada sistem kontinu atau model variabel-variabel diskrit bersamaan dengan persamaan beda atau diferensial. Sebagai tambahan, kontrol tidak sesuai dengan kejadian dari suatu peristiwa yang mungkin sebagai pemicu

menuju sistem *deadlock*, kapasitas berlebihan, atau menurunkan tingkatan kinerja sistem. Tipe dari sistem ini disebut *Discrete Event Systems* (DES) atau sering juga disebut *Discrete Event Dynamic Systems* (DEDS).

Dalam penelitian ini akan dikaji permasalahan sistem manufaktur yang dikonstruksi, dan dengan menggunakan *hybrid* sintesis Petri net untuk mendapatkan perilaku serta karakteristik sistem yang diharapkan mampu menghindari *deadlock* dalam sistem tersebut. Digunakan juga *tool* Petri net *Platform Independent Petri net Editor* (PIPE) karena menggunakan representasi grafik sehingga Pembuatan Petri net lebih mudah.

TINJAUAN PUSTAKA

PETRI NET DAN SISTEM MANUFAKTUR

Petri net diperkenalkan oleh Carl A. Petri pada tahun 1962. Sejak saat itu berkembang sebagai suatu *tool* untuk memodelkan, menganalisis dan sebagai kontrol suatu sistem *event* diskrit.

Definisi 1. (Zhou dan DiCesare, 1993) Petri net.

Petri net ditulis $Z = (P, T, I, O, m_0)$ dimana,

- P : himpunan *place-place*, secara grafik digambarkan sebagai bulatan;
- T : himpunan transisi-transisi, secara grafik digambarkan sebagai batangan, dengan $P \cup T \neq \emptyset$ dan $P \cap T = \emptyset$;
- $I: P \times T \rightarrow \{0,1\}$ adalah fungsi input secara spesifik *arc* yang diarahkan dari *place* ke transisi;
- $O: P \times T \rightarrow \{0,1\}$ adalah fungsi output secara spesifik *arc* yang diarahkan dari transisi ke *place*;
- $m: P \rightarrow T$ adalah penandaan sebanyak i komponen yaitu jumlah token, secara grafik digambarkan sebagai titik, dalam *place* ke- i . m_0 adalah inisialisasi penandaan dimana $N = \{0, 1, 2, \dots\}$.

Petri net memiliki arah, bobot, dan merupakan *graph bipartite* dengan dua bentuk *node* yaitu *place* dan transisi. *arc* (Panah) terkoneksi antara *place* dan transisi atau transisi dan *place* dan tidak akan bisa terkoneksi antar *place* atau antar transisi.

Pada bagian *place* berisi sejumlah token yang merupakan bilangan integer positif. Distribusi token pada petri net disebut penandaan (*marking*). Ketika perilaku dari suatu sistem dimodelkan dalam bentuk Petri net, penandaan diindikasikan sebagai *state* dalam suatu sistem.

Manufaktur merupakan himpunan aktivitas yang berinteraksi dengan himpunan *resource* dan hasil dalam produksi. Aktivitas yang dimaksudkan adalah proses manufaktur yang melibatkan pengerjaan dengan mesin, penanganan bahan, dan informasi pemrosesan yang harus terjadi untuk memproduksi sesuatu. *Resourcenya* merupakan personalia, mesin, bahan baku dan seterusnya yang memerlukan pelaksanaan aktivitas ini. Perencanaan proses produksi memerlukan aktivitas dan *resource* yang sangat detail. Ini meliputi prioritas yang lebih tinggi antara aktivitas, dalam artian beberapa aktivitas harus terjadi sebelum yang lain. Proses perencanaan mungkin menetapkan beberapa *resource* untuk aktivitas dan memberikan beberapa prioritas ke mana *resource* akan memilikinya.

Petri net terdefinisi dalam bentuk grafik dan matematik serta tidak memiliki arti fisik. Dalam penerapannya dibutuhkan interpretasi dengan pelabelan terhadap *place* dan transisi. Secara umum dalam pemodelan Petri net, *place* (bulat) menyatakan kondisi dan *transisi* (bar) menyatakan *event*. Dalam penelitian ini untuk memodelkan sistem manufaktur dengan menggunakan Petri net, interpretasi untuk *place*, transisi, dan token adalah sebagai berikut:

- a. *Place* merepresentasikan status *resource* atau operasi, saat mewakili bentuk pertama, jumlah inisial token adalah konstan, misalnya jumlah mesin

mengasumsikan bahwa *plantnya* adalah tetap, atau variabel, misalnya jumlah *job* atau palet pada sistem.

- b. Jika *place* merepresentasikan status *resource*, satu atau lebih token pada *place* mengartikan bahwa *resource* tersedia, dan jika tidak ada token mengartikan tidak tersedia. Jika *place* merepresentasikan operasi, maka token didalamnya menunjukkan operasi dilakukan dan jika tidak ada token mengartikan operasi tidak sedang dilakukan.
- c. Transisi merepresentasikan mulai atau selesainya suatu *event* atau penyelesaian suatu proses operasi.

MODEL KONSTRUKSI

AUTOMATED MANUFACTURING SYSTEM

Terdapat *automated manufacturing system* (AGV) yang terdiri atas komponen utama sebagai berikut: dua entri, dua exit, lima mesin, dua buffer, empat robot, dan dua sistem AGV seperti dalam Gambar 1. Berikutnya dijelaskan deskripsi masing-masing komponen utama AGV sebagai berikut:

Entri : Terdapat dua entri untuk dua tipe bahan baku, F dan G , yang memproduksi dua bentuk produksi berbeda. Setiap potongan bahan baku sudah di patenkan sehingga dapat dikirim menggunakan robot dan dua sistem AGV.

Exit: Terdapat dua exit untuk pengiriman akhir bagian F dan bagian G berturut-turut. Proses output dapat diselesaikan via dua sistem AGV.

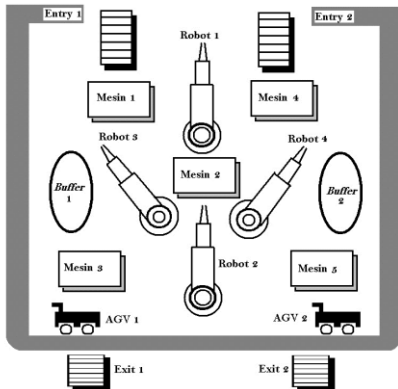
Mesin: Mesin M_1 dan M_2 memproses potongan bahan F dari entri 1. Kedua mesin memproduksi bagian F yang nantinya akan dikirim ke *buffer* 1. Bagian perantara ini selanjutnya diproses oleh M_3 untuk memproduksi bagian akhir dari bahan F . Mesin M_4 memproses potongan bahan dari entri 2 dan memproduksi bagian-bagian G yang akan dikirim ke *buffer* 2. Mesin M_5 merupakan perantara dari bahan G dari *buffer* 2.

Buffer: *buffer* 1 pada ukuran b_1 mendapatkan bagian F dari M_1 atau M_2 . Output ini adalah perantara bagian-bagian F satu per satu ke M_3 . *buffer* 2 pada ukuran b_2 mendapatkan proses bagian G dari M_4 . Output ini merupakan bagian-bagian G satu per satu ke M_5 .

Robot: Robot R_1 mengisi M_1 , M_2 dan M_3 , dan tidak dari ketiga mesin merupakan prioritas untuk dilayani robot tersebut. Dalam aturan *non-deterministik*, R_1 memilih mesin jika bahan baku sudah tersedia dan mesin sudah siap. Robot R_2 membongkar M_3 dan M_5 , serta mengirim bagian-bagian yang sudah selesai/lengkap ke palet yang ada pada sistem AGV. Robot R_3 berbagi dengan M_1 , M_2 dan M_3 . Berikut fungsi yang dilaksanakan : membongkar M_1 , dan M_2 sampai bagian perantara tipe F ke dan dari *buffer* 1, dan mengisi M_3 . Setiap pemrosesan bagian F

dengan M_1 atau M_2 harus disimpan dalam *buffer* 1 sebelum proses selanjutnya.

Desain seperti ini mencegah pemrosesan yang baru berjalan oleh M_1 atau M_2 dari proses perantara oleh M_3 . Sama halnya, robot R_4 digunakan untuk melayani M_4 dan M_5 . Ini bisa digunakan untuk membongkar M_4 untuk mengirim bagian perantara tipe F ke dan dari *buffer* 2, dan mengisi M_5 . *Buffer* 2 menjalankan fungsi yang sama seperti yang dilakukan *buffer* 1. **Sistem AGV:** Dua AGV didesain untuk pengiriman bahan final dan menjalankan palet dalam sistem.



Gambar 1. Stasiun Transfer Bahan baku

Dari M_3 , AGV 1 mengirim bagian final F ke output dan palet kembali ke entri 1 dari M_5 , AGV 2 mengirim bahan final G ke output dan palet kembali ke entri 2. Saat bagian berbeda diambil *deadlock* dapat dihindari. Baik keduanya, AGV dapat bekerja bersamaan setelah robot 2 meletakkan bagian pada palet satu per satu.

METODE SINTESIS

Metode sintesis terdiri dari dua bagian utama, yang masing-masing memiliki bagian penting dalam mendesain sistem. Metode sintesis terbagi menjadi desain *top-down* dan desain *bottom-up*. Berikut deskripsi dari metode sintesis.

Desain Top-down: Langkah 1: memilih Petri net yang *reversible*, *live*, dan *bounded* sebagai model *first-level* dari suatu sistem yang bekerja saat semua *resource* utama tersedia. dan menentukan P_A , P_B , dan P_C dimana P_A dan P_B merupakan subjek untuk meningkatkan, sementara P_C selalu tetap dan anggotanya memiliki jumlah variabel pada inisial token. Langkah 2: menguraikan sistem dalam beberapa subsistem di perlihatkan sebagai operasi *place* menggunakan modul desain dasar terdefinisi. Gantikan operasi *place* ini dengan modul desain dasar yang semakin detail sampai tidak ada operasi-operasi yang dapat dibagi selanjutnya atau sampai pada titik jangkauan yang tidak diperlukan. P_A semakin meningkat di setiap langkah. Langkah 3: secara wajar menambah *place non-shared resource* pada setiap langkah, saat satu dari

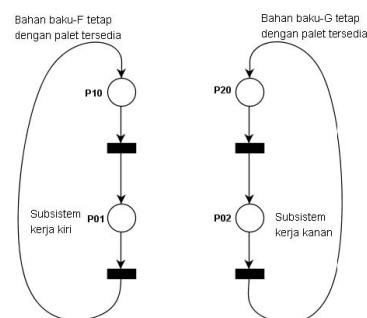
beberapa operasi memerlukan *resource*. *Arc* terhubung ke *place resource* sehingga Akibat 1 terpenuhi. Pada setiap penambahan hanya P_B yang ditingkatkan.

Desain Bottom-up: Langkah 4: menambahkan setiap *place buffer*, yang dipasangkan dengan *place-A* yang sudah ada untuk merepresentasikan *buffer*, bentuk modul *buffer* Petri net jika setiap *buffer* ada. Setiap penambahan *place buffer* dengan inialisasi token yang mana jumlahnya mengartikan kapasitas pada *buffer* yang menjadi *place-B*. Langkah 5: menambahkan *place shared resource* yang merupakan bentuk *Paralel Mutual Exclusion* (PME) atau *Generalized Paralel Mutual Exclusion* (GPME). *Place-place resource* akan menjadi *place-B*. Langkah 6: menambahkan *place shared resource* yang merupakan bentuk *Sequential Mutual Exclusion* (SME) atau *Generalized Sequential Mutual Exclusion* (GSME) dan menghitung kapasitas token yang terjangkau antara PME atau GPME, α , β , dan γ , serta menentukan jumlah inisial token dalam $p, \forall p \in P_B$.

HASIL DAN PEMBAHASAN

Pada bagian ini akan di bahas proses modeling Petri net dengan menggunakan deskripsi sintesis.

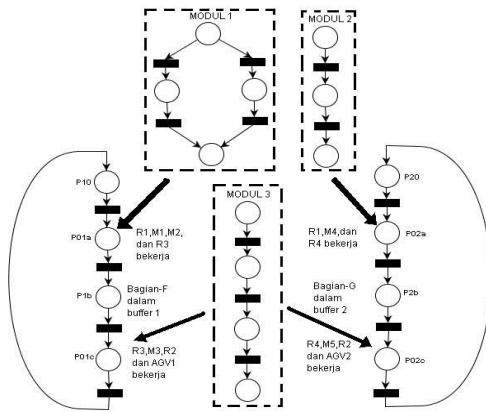
Desain Petri net level pertama: tanpa mempertimbangkan *shared resource*, robot, sistem ini dapat ditunjukkan dalam dua subsistem yang saling bergantung subsistem kiri dan subsistem kanan. Subsistem kiri memuat tiga mesin, *buffer*, dan AGV. Subsistem kanan memuat dua mesin yang terhubung, *buffer* dan AGV. Saat struktur *choice-synchronization* tidak dilibatkan dalam dua subsistem ini, Petri net level pertama didesain dengan model yang memiliki dua bagian Petri net seperti pada Gambar 2. Kedua net merupakan *choice-free net*. Secara inisial ditulis: $P_A = \{p_{01}, p_{02}\}$; $P_B = \emptyset$; $P_C = \{p_{10}, p_{20}\}$.



Gambar 2. Model Petri net level pertama memuat dua bagian net

Dekomposisi dan penghalusan operasi: untuk menjabarkan sistem ini secara lengkap, dekomposisi dan penghalusan sangat diperlukan. Kedua subsistem diuraikan menjadi tiga langkah untuk *generate* produk akhir, baik p_{01} dan p_{02} pada model Petri net level pertama dalam Gambar 3 digantikan dengan tiga modul barisan *place*. Proses penghalusan net diperoleh sesuai Gambar 3 dimana,

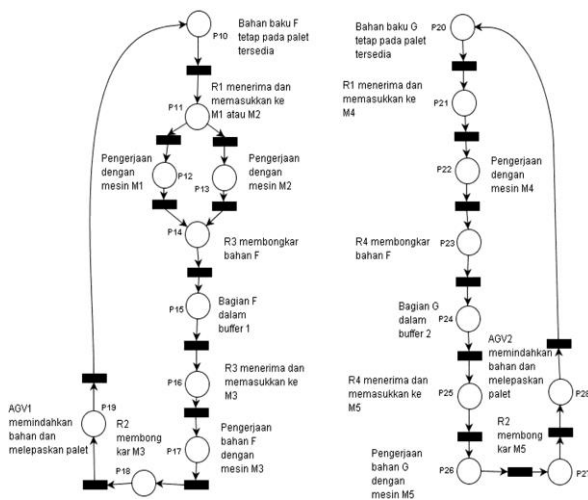
$$P_A = \{p_{01a}, p_{1b}, p_{01c}, p_{02a}, p_{2b}, p_{02c}\}.$$



Gambar 3. Petri net setelah penghalusan pada p_{01} dan p_{02}

Arti masing-masing *place* dijelaskan pada Gambar. 4 dan Tabel. 1. Selanjutnya dekomposisi dan penghalusan dilakukan yaitu:

1. *Place* p_{01a} diganti dengan modul PN pilihan (modul 1 pada Gambar 3)
2. *Place* p_{02a} diganti dengan tiga barisan *place* modul PN (modul 2 pada Gambar 3)
3. Baik p_{01c} dan p_{02c} diganti dengan empat barisan *place* modul PN (modul 3 pada Gambar 3)
4. *Place* p_{01b} dan p_{02b} model perantara bagian *F* dan *G*, masing-masing secara berurutan disimpan pada *buffer* 1 dan *buffer* 2. Setelah itu penghalusan untuk kedua *place* ini tidak dibutuhkan.



Gambar 4. Petri net setelah penghalusan pada p_{01a} , p_{02a} , p_{01c} dan p_{02c}

Tabel 1. Label *Place* untuk Petri net pada Gambar 3.

Place	Interpretasi
p_{10}	Bahan baku tipe <i>F</i> tersedia, Masing-masing palet siap
p_{01a}	R_1, M_1, M_2 , dan R_3 bekerja
p_{1b}	Bagian <i>F</i> dalam <i>buffer</i> 1
p_{01c}	R_3, M_3, R_2 , dan AGV1 bekerja
p_{20}	Bahan baku tipe <i>G</i> , Masing-masing tetap dengan palet tersedia
p_{02a}	R_1, M_4 , dan R_4 bekerja
p_{2b}	Bagian <i>G</i> dalam <i>buffer</i> 2
p_{02c}	R_4, M_5, R_2 , dan AGV2 bekerja

Selanjutnya akan ditunjukkan penghalusan tersebut dengan melibatkan operasi *place* pada Gambar 5. Himpunan *place* operasi adalah, $P_A = p_{ij}, p_{19}, i = 1,2$, dan $1 \leq j \leq 8$. Interpretasi *place* ini ditunjukkan pada Tabel 2.

Tabel 2. Label *Place* untuk Petri net pada Gambar 4.

Place	Interpretasi
p_{10}	Bahan baku tipe <i>F</i> tersedia, Masing-masing palet siap
p_{11}	R_1 memperoleh palet dari entri 1 dan mengisi M_1 atau M_2
p_{12}	Pengerjaan dengan mesin M_1 bagian bahan <i>F</i>
p_{13}	Pengerjaan dengan mesin M_2 bagian bahan <i>F</i>
p_{14}	R_3 membongkar M_1 atau M_2 dan menempatkan bagian <i>F</i> pada <i>buffer</i> 1
p_{15}	<i>Buffer</i> 1 menyimpan bagian <i>F</i>
p_{16}	R_3 memperoleh palet dari <i>buffer</i> 1 dan mengisi M_3
p_{17}	Pengerjaan dengan mesin M_3 bagian bahan <i>F</i>
p_{18}	R_2 membongkar M_3 dan meletakkan palet pada AGV 1
p_{19}	AGV 1 menyelesaikan bagian <i>F</i> ke exit 1 dan melepaskan palet ke entri 1
p_{20}	Bahan baku tipe <i>G</i> tersedia, Masing-masing palet siap
p_{21}	R_1 memperoleh palet dari entri 2 dan mengisi M_4
p_{22}	Pengerjaan dengan mesin M_4 bagian bahan <i>G</i>
p_{23}	R_4 membongkar M_4 dan menempatkan bagian <i>G</i> pada <i>buffer</i> 1
p_{24}	<i>Buffer</i> 2 menyimpan bagian <i>G</i>
p_{25}	R_4 memperoleh palet dari <i>buffer</i> 2 dan mengisi M_5
p_{26}	Pengerjaan dengan mesin M_5 bagian bahan <i>G</i>
p_{27}	R_2 membongkar M_5 dan meletakkan palet pada AGV 2
p_{28}	AGV 2 menyelesaikan bagian <i>G</i> ke exit 2 dan melepaskan palet ke entri 2

Penambahan *Non-Shared Resource*: Lima mesin dan dua AGV digunakan dengan proses operasi tunggal, sehingga merupakan *resource non-shared* dalam sistem. akan ditambahkan *place resource* pada Petri net. Semua penambahan *place resource* ditunjukkan pada Gambar 5. Penambahan *place* baru merupakan milik himpunan *place resource* tetap.

$$P_B = \{p'_{12}, p'_{13}, p'_{17}, p'_{19}, p'_{22}, p'_{26}, p'_{28}\}$$

Penjelasan atas *place-B* dijelaskan pada Tabel 3.

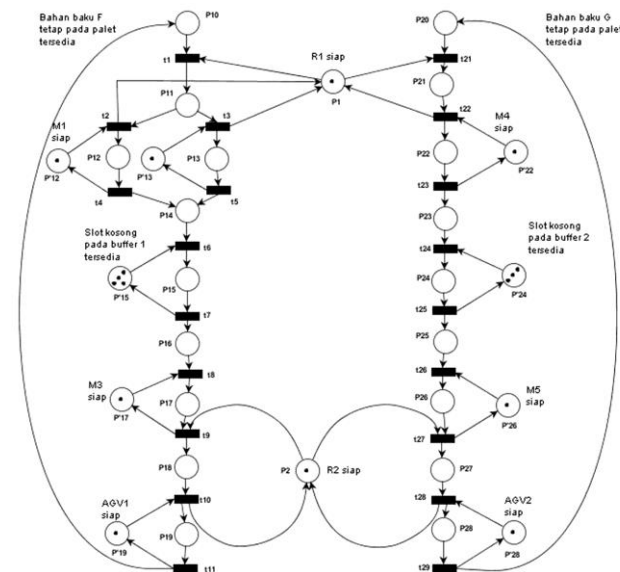
Penambahan *place buffer*: Sistem ini terdiri dari dua *buffer* sederhana. Dua *place-B* dapat diartikan sebagai, p'_{15} merupakan slot kosong dalam *buffer* 1 tersedia jika terdapat token di dalamnya dan p'_{24} merupakan slot kosong dalam *buffer* 2 yang juga tersedia jika terdapat token di dalamnya. Khususnya, p'_{15}, p_{15} dan dua transisinya memenuhi kondisi pertama pada teorema yaitu

Z' bounded (safe) jika Z bounded (safe), konstruksi ini menghasilkan modul buffer yang menjaga sifat net sehingga boundedness, liveness, dan reversibility. Argumen yang sama dapat diterapkan untuk p'_{24} dan kedua transisinya. Ini jelas bahwa $m_0(p'_{15}) = b_1$ dan $m_0(p'_{16}) = b_2$.

Tabel 3. Label place untuk Petri net setelah penambahan non-shared resource

Place	Interpretasi
p'_{12}	M_1 siap
p'_{13}	M_2 siap
p'_{17}	M_3 siap
p'_{19}	AGV 1 siap
p'_{22}	M_4 siap
p'_{26}	M_5 siap
p'_{28}	AGV 2 siap

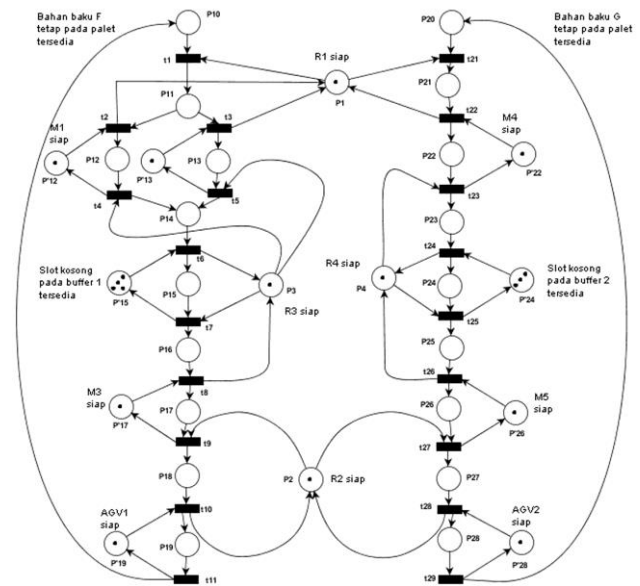
Penambahan shared resource untuk membentuk GPME: dengan mempertimbangkan robot R_2 yang berbagi dengan dua mesin M_3 dan M_5 . Setelah p_2 dan arcnya ditambahkan ditambahkan seperti terlihat pada Gambar 5, nyata bahwa $(p_2, (t_9, t_{10}), (t_{27}, t_{28}))$ adalah PME-2. Oleh sebab itu, net bounded, live, dan reversible. Selanjutnya dengan mempertimbangkan robot R_1 berbagi dengan M_1 , M_2 dan M_4 . Saat M_1 dan M_2 berbagi dengan entri, setelah R_1



Gambar 5. Petri net setelah penambahan Robot R_1 dan R_2 memperoleh bahan baku F

Selanjutnya dapat mengisi M_1 atau M_2 saat keduanya idle. Share resource ini dimodelkan dalam Gambar 5. menggunakan place-B p_1 . Hal ini juga memudahkan untuk memverifikasi bahwa $(p_1, \{(t_1, \{t_2, t_3\}), (t_{21}, t_{22})\})$ membentuk GPME. Oleh sebab itu teori menjamin boundedness, liveness, dan reversibility pada net dalam Gambar 5. selama token berada pada place-C. Dua place-B, p_1 dan p_2 , masing-masingnya berisi satu token, dengan keterangan p_1 : R_1 siap dan p_2 : R_2 siap.

Penambahan shared resource untuk membentuk GSME: bagian yang paling sulit adalah melakukan pencocokan dengan barisan shared resource. Sistem ini menyediakan dua macam resource, robot R_3 dan R_4 . Sebagai contoh, R_4 pertama dibutuhkan untuk membongkar M_4 , dan kemudian mungkin membongkar M_4 lagi atau M_5 . membongkar M_4 dan M_5 merupakan dua barisan proses yang saling terkait. Selanjutnya hasil akhir desain Petri net digambarkan pada Gambar 6. Proses pemodelan dapat dibagi atas beberapa tahapan. Pertama, analisa robot R_3 . Saat hanya arc $(p_3, t_4), (p_3, t_5)$ dan (t_6, p_3) ditambahkan pada net, maka GPME-1 terbentuk yaitu, $(p_3, \{(t_4, t_5), t_6\})$. Saat arc (p_3, t_7) dan (t_8, p_3) ditambahkan, maka struktur $(p_3, \{(t_4, t_5), t_6\}) \cup \{(t_7, t_8)\}$ dapat diverifikasi sebagai GSME untuk sistem.



Gambar 6. Model akhir Petri net untuk AMS

Selanjutnya analisis kapasitas token, α , β , dan γ adalah sebagai berikut:

$$C^1 = C(t_6, t_7) = b_1.$$

$$\alpha(p_{10}, t_4) = \alpha(p_{10}, t_5) = 1, \alpha(p_{20}, t_4) = \alpha(p_{20}, t_5) = 0;$$

$$\alpha(p_{10}) = 1, \alpha(p_{20}) = 0.$$

$$\beta(p_{10}, t_4) = \beta(p_{10}, t_5) = 1, \beta(p_{20}, t_4) = \beta(p_{20}, t_5) = 0;$$

$$\beta(p_{10}) = 1, \beta(p_{20}) = 0;$$

$$\gamma(p_{10}, t_4) = \gamma(p_{10}, t_5) = m_0(p_{10}).$$

Sedemikian sehingga, dengan menggunakan teorema untuk GSME, inisial penandaan yang mungkin adalah:

$$1 \leq m_0(p_{10}) \leq b_1,$$

dan

$$1 \leq m_0(p_{20})$$

yang diperlukan untuk liveness dan reversibility. Kedua, analisa robot R_4 . Struktur $(p_4, \{t_{23}, t_{24}\}) \cup \{(t_{25}, t_{26})\}$ dapat diverifikasi sebagai SME. Kapasitas token, α , β , dan γ dapat dihitung sebagai berikut:

$$C^1 = C(t_{24}, t_{25}) = b_2.$$

$$\alpha(p_{10}) = 0, \alpha(p_{20}) = 1$$

$$\beta(p_{10}) = 0, \beta(p_{20}) = 1$$

dan

$$\gamma(p_{20}, t_{23}) = m_0(p_{20})$$

demikian sehingga,

$$\gamma(t_{23}) = \infty$$

Dengan menggunakan teorema untuk SME, diperoleh:

$$1 \leq m_0(p_{10})$$

Sedemikian sehingga, dengan tujuan untuk menjamin sifat kualitatif yang diharapkan pada net seperti Gambar 6, inisial penandaan harus memenuhi:

$$1 \leq m_0(p_{10}) \leq b_1,$$

dan

$$1 \leq m_0(p_{20}) \leq b_2.$$

Dengan inisial penandaan yang telah diperoleh dan struktur Petri net seperti yang ditunjukkan pada Gambar 6, maka Petri net *bounded*, *live*, dan *reversible*. Himpunan *place* operasi, himpunan *place fixed resource*, dan himpunan variabel *resource* dapat ditulis sebagai berikut:

$$P_A = \{p_{1i}, 1 \leq i \leq 9\} \cup \{p_{2i}, 1 \leq i \leq 8\}$$

$$P_B = \{p_1, p_2, p_3, p_4, p'_{12}, p'_{13}, p'_{15}, p'_{17}, p'_{19}, p'_{22}, p'_{24}, p'_{26}, p'_{28}\}$$

dan

$$P_C = \{p_{10}, p_{20}\}$$

Sedangkan dengan tujuan untuk mengurangi kemungkinan terjadinya pada lingkungan operasional, dipilih:

$$m_0(p_{10}) = b_1$$

dan

$$m_0(p_{20}) = b_2$$

KESIMPULAN

Kesimpulan yang dapat diambil adalah dengan menggunakan metode *hybrid*, Petri net untuk sistem manufaktur dapat disintesis satu per satu, sehingga menghasilkan Petri net *bounded*, *live* dan *reversible* dengan 32 *place* dan 20 transisi.

DAFTAR PUSTAKA

- Adzkiya, D. 2008, *Membangun Petri Net Lampu Lalu Lintas dan Simulasinya*, Tesis Magister, Institut Teknologi Sepuluh Nopember, Surabaya.
- David, R. dan Alla, H. 2005, *Discrete, Continuous, and Hybrid Petri Nets*, Springer-Verlag Berlin Heidelberg, New York.
- Murata, T. 1989, *Petri Net: Properties, Analysis, and Applications*, Proceedings of The IEEE, Eds: Department of Electrical Engineering and Computer Science, University of Illinois, Chicago, Vol. 77, No. 4, hal. 541-580.
- Zhou, M. dan DiCesare, F. 1993. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic Publishers, Massachusetts.