

# PENERAPAN PETRI-NET PADA MODEL GERAKAN BERJALAN *WALKING* ROBOT BERKAKI EMPAT (*QUADRUPED*)

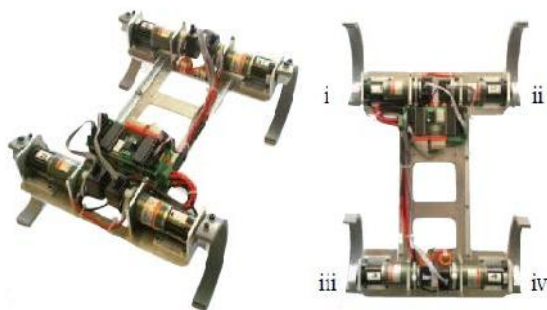
Kuzairi, Toni Yulianto, M. Fariz Mfadillah Mardianto, Faisol, Rica Amalia  
Jurusan Matematika, Fakultas MIPA, Universitas Islam Madura (UIM)  
Kampus Hijau Komplek PP. Miftahul Ulum Bettet, Pamekasan 69351  
e-mail: Kuzairi81@gmail.com

**Abstrak**—Selama ini sudah banyak robot diproduksi baik dalam skala kecil maupun besar untuk membantu manusia dalam mengerjakan pekerjaan sehari-hari manusia sesuai dengan fungsi masing-masing. Akan tetapi, fungsi-fungsi tersebut akan dapat berjalan dengan baik apabila komponen-komponen yang mendukung pada robot dapat berjalan dengan baik, seperti salah satunya adalah cara gerak berjalan robot. Di sini, peneliti lebih menekankan gerak jalan robot empat kaki yang menggunakan *walking*, karena yang banyak umum digunakan dan kesulitannya juga lumayan dibandingkan robot berjalan dengan dua kaki. Maka dengan menggunakan aljabar max plus akan diperoleh model gerak jalan robot *walking* tersebut yang sesuai dengan yang diharapkan.

**Kata Kunci**—Aljabar Max Plus, gerak jalan *walking*, Petri net, robot empat kaki.

## I. PENDAHULUAN

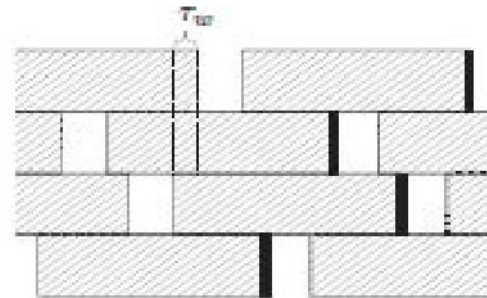
Pada gerak robot berjalan dari sebuah robot berkaki dua biasa (kontinu), kaki kiri hanya boleh terangkat (*lift off*) dari tanah setelah kaki kanan mendarat (*touched down*), untuk memastikan robot tidakjatuh karena kurangnya penyangga (*support*). Keperluan sinkronisasi ini dapat dimodelkan dengan evolusi dari sistem kejadian diskrit dengan meringkaskan setiap siklus batas pada lingkaran ke dalam dua peristiwa berurutan dalam sirkuit tertutup [5]: lepas landas (*lift off*) dan mendarat (*touchdown*). Tiap kaki kemudian dimodelkan oleh sebuah siklus kejadian dan sinkronisasi antara kaki diberlakukan oleh koneksi antara tiap siklus kaki yang digambarkan dalam Gambar 1.1. Sinkronisasi dari beberapa kaki adalah sangat penting khususnya untuk robot pendaki, dimana kurangnya penyangga dapat mengakibatkan konsekuensi yang berakibat fatal.



Gambar 1.1. Sudut pandang dan pandangan dari atas pada robot berkaki empat yang dikembangkan di DCSC, TU Delft, terinspirasi oleh robot *hexapodRhex*. Nomor romawi pada gambar kanan merepresentasikan urutan penomoran kaki

## II. CARA BERJALAN *WALKING* ROBOT BERKAKI EMPAT (*QUADRUPED GAITS*)

Gambar 2.1 mengilustrasikan salah satu cara berjalan untuk quadruped yakni jalan pelan (*paceing*) [3]. *Pacing*, umumnya digunakan oleh unta dan unta betina, merupakan cara berjalan bilateral, yaitu, kaki kiri depan dan belakang disinkronkan dan berlawanan fase dengan kaki kanan depan dan belakang.



Gambar 2.1. Evolusi waktu dari cara berjalan untuk *quadruped*. Kotak yang berarsir merepresentasikan posisi kaki ketika diam di tanah (*stance*), dan garis tebal padat merepresentasikan kejadian *lift off*<sub>*i*</sub>(*k* + 1). Panah pada a) merepresentasikan kejadian yang harus berlangsung untuk setiap kejadian *lift off* dan *touchdown* agar terjadi.

Keterangan dari Gambar 2.1:

- $t_i(k)$  Waktu *touchdown* untuk kaki ke-*i* pada iterasi ke-*k*
- $l_i(k)$  Waktu *lift off* untuk kaki ke-*i* pada iterasi ke-*k*
- i* Indeks untuk kaki
- $\tau_f$  Waktu yang dibutuhkan oleh kaki ketika berada di udara
- $\tau_g$  Waktu yang dibutuhkan oleh kaki ketika berada di tanah
- $\tau_w$  Parameter pergantian waktu untuk cara berjalan *walking*

## III. GRAF KEJADIAN DENGAN WAKTU

Teori Petri net menyediakan alat intuitif untuk mewakili DES (*Discrete Event System*) secara grafis yang dinamakan graf Petri net [5]. Hal ini mudah untuk menghasilkan cara berjalan dengan mengembangkan suatu Petri net dalam waktu jika: 1) tiap kaki dimodelkan sebagai sebuah sirkuit dengan transisi *lift off* dan *touchdown* 2) sinkronisasi antar kaki diimplementasikan dengan melakukan *fire* kondisi pada transisi *lift off*. Selain itu, dengan merancang Petri net dapat diambil keuntungan dalam hal alat analisis dan sintesis aljabar max-plus. Sebuah sub kelas Petri net dengan waktu yang disebut graf kejadian secara resmi setara dengan sistem

*linearmax-plus* [5]. Hal ini memungkinkan untuk desain intuitif dari cara berjalan oleh graf kejadian dengan terjemahan langsung ke sistem *linear max-plus* dan sebaliknya [4].

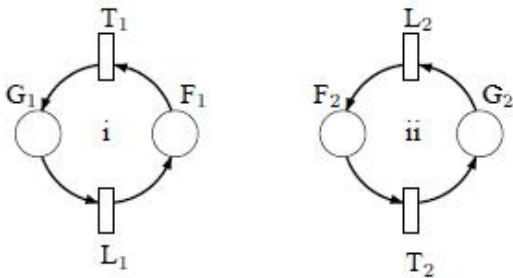
**Definisi 3.1**

Sebuah Petri net dengan waktu  $\mathcal{G}$  dikarakterisasikan oleh sebuah himpunan *place*  $\mathcal{P}$ , sebuah himpunan transisi  $\mathcal{Q}$ , sebuah himpunan garis  $\mathcal{D}$  dari transisi ke *place* dan sebaliknya, sebuah tanda awal (*initial marking*)  $\mathcal{M}_0$ , dan vektor pemegang waktu  $\mathcal{T}$ . Jika tiap *place* memiliki tepat satu transisi *upstream* dan satu transisi *downstream*, maka Petri net dengan waktu disebut graf kejadian dengan waktu (*timed event graph*).

Dari Gambar 2.1 terlihat bahwa kaki *i*, *ii*, *iii*, dan *iv* memiliki waktu *lift off* dan *touchdown* yang berbeda-beda. Oleh karena itu, pada cara berjalan *walking* ini kaki tidak memiliki waktu kejadian yang sama, sehingga masing-masing kaki bekerja sendiri-sendiri. Asumsikan tiga kaki berada di tanah bersama-sama.

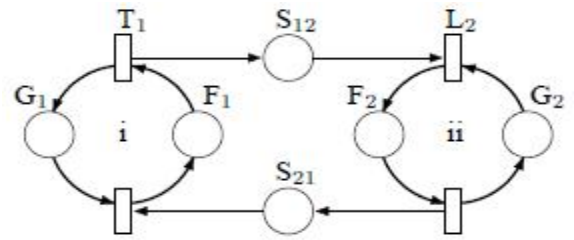
Dengan konstruksi graf kejadian dari cara berjalan *walking* dihasilkan dari waktu dengan cara berjalan *pacing* adalah sebagai berikut [2]:

- 1) Untuk tiap kaki, definisikan sebuah sirkuit dengan dua peristiwa (*event*): *touchdown*  $T_i$  dan *lift off*  $L_i$ . Antar peristiwa, tambahkan *place*  $G_i$  untuk waktu ketika kaki *i* berada di tanah dan  $F_i$  untuk saat kaki berada di udara. Hal ini ditunjukkan oleh dua sirkuit tebal bernomor seperti diilustrasikan pada Gambar 3.1:



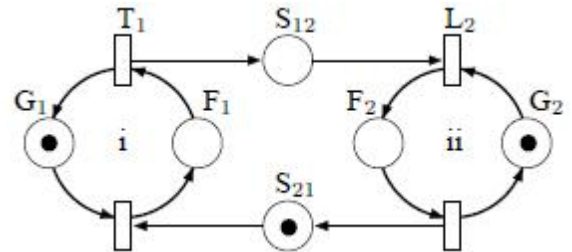
Gambar 3.1. *place*  $G_i$  untuk waktu ketika kaki *i* berada di tanah dan  $F_i$  untuk saat kaki berada di udara.

- 2) Untuk tiap sinkronisasi yang dibutuhkan, tambahkan sebuah *place* antara transisi. Hal ini penting untuk memastikan bahwa tiap *place* memiliki sebuah garis *upstream* yang unik dan sebuah garis *downstream* yang unik. Sebagai contoh, *place* berlabel  $S_{12}$  menentukan bahwa transisi *lift off* kaki '*ii*' hanya boleh di-*fire* setelah kaki '*i*' telah menyentuh tanah. Hal ini diilustrasikan oleh Gambar 3.2:



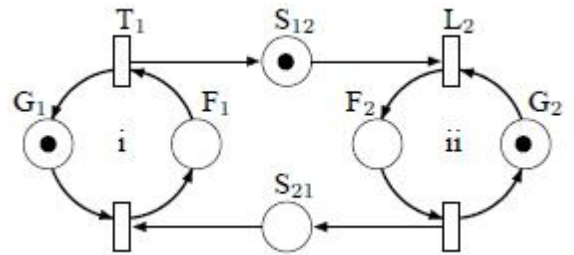
Gambar 3.2. *place* antara transisi.

- 3) Inisialisasi tanda sedemikian hingga semua *place* yang menyatakan kaki sedang berada di tanah (*ground place*) memiliki sebuah token (robot memulai gerakan dengan semua kaki berada di tanah). Selain itu, token ditambahkan pada *place* tersinkronisasi sedemikian rupa sehingga semua sirkuit (tertutup) dapat hidup dan tanda awal dapat dilaksanakan. Hal ini dapat diilustrasikan oleh Gambar 3.3.a dan 3.3.b:



Gambar 3.3.a. Inisialisasi tanda jika  $S_{21}$  dilakukan.

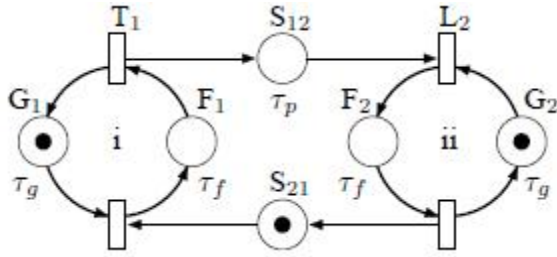
jika kaki yang akan terangkat terlebih dahulu adalah kaki '*i*' dan



Gambar 3.3.b Inisialisasi tanda jika  $S_{12}$  dilakukan.

jika kaki yang akan terangkat terlebih dahulu adalah kaki '*ii*'.

Jadi secara lengkapnya, graf kejadian dengan waktu atau Petri net dengan waktu untuk cara berjalan *walking* yang bersumber dari cara berjalan *pacing* dapat digambarkan pada Gambar 3.4:



Gambar 3.4. Graf kejadian dengan waktu untuk cara berjalan *pacing*

#### IV. MODEL ALJABAR MAX-PLUS PADA PETRI-NET CARA BERJALAN *WALKING*

Mengikuti langkah-langkah yang disajikan di atas, adalah mungkin untuk mereproduksi banyak cara berjalan. Setelah graf kejadian dengan waktu dibangun, akan sangat mudah untuk menemukan sistem *linear max-plus* yang terkait [4]:  $\kappa_j \Psi_i \psi_i v_j$

- 1) Tiap transisi  $\Psi_i$  dalam graf kejadian dengan waktu diberikan ke sebuah variabel kejadian  $\psi_i$  dalam aljabar *max-plus*.
- 2) Untuk tiap garis masuk ke transisi  $\Psi_i$  buat daftar dari
  - Kejadian  $\Psi_j$  yang mendahului kejadian  $\Psi_i$  untuk garis tersebut dan tambahkan  $j$  ke dalam suatu himpunan  $\mathcal{S}$
  - Konstanta waktu  $v_j$  dari *place* asal garis ini
  - jumlah token  $\kappa_j$  dari *place* asal garis ini

Kemudian tulis ekspresi:  $T_i$

$$\psi_i(k+1) = \max_{j \in \mathcal{S}} (\psi_j(k+1 - \kappa_j) + v_j) \quad (4.1)$$

Sebagai contoh, lihat transisi  $T_1$  pada Gambar 3.1. Transisi ini memiliki garis masuk tunggal dari transisi  $L_1$  dengan konstanta waktu  $\tau_f$  dan nol token. Dengan demikian, persamaan (4.1) untuk transisi ini adalah

$$t_1(k+1) = l_1(k+1) + \tau_f$$

Untuk transisi  $L_1$  terdapat dua garis masuk dari  $T_1$  dan  $T_2$ , dengan masing-masing konstanta waktu  $(\tau_g, \tau_p)$  dan satu token per *place*. Maka persamaan transisi untuk  $l_1$  adalah:

$$l_1(k+1) = \max(t_1(k) + \tau_g, t_2(k) + \tau_p)$$

Mengikuti prosedur ini maka diperoleh persamaan transisi untuk cara berjalan *pacing* yakni:

$$t_1(k+1) = l_1(k+1) + \tau_f$$

$$l_1(k+1) = \max(t_1(k) + \tau_g, t_2(k) + \tau_p)$$

$$t_2(k+1) = l_2(k+1) + \tau_f$$

$$l_2(k+1) = \max(t_2(k) + \tau_g, t_1(k+1) + \tau_p) \quad (4.2)$$

Himpunan persamaan (4.2) dapat diubah ke dalam struktur persamaan  $x(k+1) = A_{(m(k))} \otimes x(k)$ , dengan substitusi rekursif [3]. Sebagai contoh, persamaan (4.2) diupdate untuk  $t_1$  menjadi:

$$t_1(k+1) = l_1(k+1) + \tau_f \\ = \max(t_1(k) + \tau_g, t_2(k) + \tau_p) + \tau_f \quad (4.3)$$

$$= \max(t_1(k) + \tau_g + \tau_f, t_2(k) + \tau_p + \tau_f)$$

Mengikuti proses rekursif yang sama pada persamaan (4.2) dan (4.3) untuk variabel lainnya pada cara berjalan *walking*, dan mendefinisikan variabel keadaan  $x(k) \in \mathbb{R}_{max}^4$  dengan  $x(k)$

$= [t_1(k) \ l_1(k) \ t_2(k) \ l_2(k) \ t_3(k) \ l_3(k) \ t_4(k) \ l_4(k)]^T$ , dapat diperoleh matriks sistem *linear-max-plus* untuk cara berjalan *walking*, yang dinotasikan  $A_w$ . Sehingga persamaan transisi untuk berjalan *walking* diperoleh [1]:

$$t_1(k+1) = l_1(k+1) + \tau_f$$

$$l_1(k+1) = \max(t_1(k) + 3\tau_{wf} + \tau_w, t_2(k) + 5\tau_{wf} + \tau_w, t_3(k) + 4\tau_{wf} + \tau_w, t_4(k) + 3\tau_{wf} + \tau_g)$$

$$t_2(k+1) = l_2(k+1) + \tau_f$$

$$l_2(k+1) = \max(t_2(k) + 3\tau_{wf} + \tau_w, t_1(k) + \tau_{wf} + \tau_w, t_3(k) + 2\tau_{wf} + \tau_w, t_4(k) + \tau_{wf} + \tau_g)$$

$$t_3(k+1) = l_3(k+1) + \tau_f$$

$$l_3(k+1) = \max(t_3(k) + 3\tau_{wf} + \tau_w, t_1(k) + 2\tau_{wf} + \tau_w, t_2(k) + 4\tau_{wf} + \tau_w, t_4(k) + 2\tau_{wf} + \tau_g)$$

$$t_4(k+1) = l_4(k+1) + \tau_f$$

$$l_4(k+1) = \max(t_4(k) + \tau_g, t_1(k) + \tau_w, t_2(k) + 2\tau_{wf} + \tau_w, t_3(k) + \tau_{wf} + \tau_w) \quad (4.4)$$

Untuk cara berjalan simetri diasumsikan bahwa

$$\tau_g > 3\tau_f, \text{ dan } \tau_w = \frac{\tau_g - 3\tau_f}{4} \quad (4.5)$$

Parameter tambahan  $\tau_{wf} = \tau_w + \tau_f$  dan sistem matriks  $A_w$  yang didapat dari persamaan (4.4) dan (4.5) diperkenalkan untuk penyederhanaan [4].

$$A_w = \begin{bmatrix} 4\tau_{wf} & \varepsilon & 6\tau_{wf} & \varepsilon & 5\tau_{wf} & \varepsilon & 3\tau_{wf} + \tau_{fg} & \varepsilon \\ 3\tau_{wf} + \tau_w & \varepsilon & 5\tau_{wf} + \tau_w & \varepsilon & 4\tau_{wf} + \tau_w & \varepsilon & 3\tau_{wf} + \tau_g & \varepsilon \\ 2\tau_{wf} & \varepsilon & 4\tau_{wf} & \varepsilon & 3\tau_{wf} & \varepsilon & \tau_{wf} + \tau_{fg} & \varepsilon \\ \tau_{wf} + \tau_w & \varepsilon & 3\tau_{wf} + \tau_w & \varepsilon & 2\tau_{wf} + \tau_w & \varepsilon & \tau_{wf} + \tau_g & \varepsilon \\ 3\tau_{wf} & \varepsilon & 5\tau_{wf} & \varepsilon & 4\tau_{wf} & \varepsilon & 2\tau_{wf} + \tau_{fg} & \varepsilon \\ 2\tau_{wf} + \tau_w & \varepsilon & 4\tau_{wf} + \tau_w & \varepsilon & 3\tau_{wf} + \tau_w & \varepsilon & 2\tau_{wf} + \tau_g & \varepsilon \\ \tau_{wf} & \varepsilon & 3\tau_{wf} & \varepsilon & 2\tau_{wf} & \varepsilon & \tau_{fg} & \varepsilon \\ \tau_w & \varepsilon & 2\tau_{wf} + \tau_w & \varepsilon & \tau_{wf} + \tau_w & \varepsilon & \tau_g & \varepsilon \end{bmatrix} \quad (4.6)$$

Dengan mengganti parameter  $\tau_g = 3s$  dan  $\tau_f = 1s$ , maka matriks  $A_w$  pada persamaan (4.6) menjadi:

$$A_w = \begin{bmatrix} 4 & \varepsilon & 6 & \varepsilon & 5 & \varepsilon & 7 & \varepsilon \\ 3 & \varepsilon & 5 & \varepsilon & 4 & \varepsilon & 6 & \varepsilon \\ 2 & \varepsilon & 4 & \varepsilon & 3 & \varepsilon & 5 & \varepsilon \\ 1 & \varepsilon & 3 & \varepsilon & 2 & \varepsilon & 4 & \varepsilon \\ 3 & \varepsilon & 5 & \varepsilon & 4 & \varepsilon & 6 & \varepsilon \\ 2 & \varepsilon & 4 & \varepsilon & 3 & \varepsilon & 5 & \varepsilon \\ 1 & \varepsilon & 3 & \varepsilon & 2 & \varepsilon & 4 & \varepsilon \\ 0 & \varepsilon & 2 & \varepsilon & 1 & \varepsilon & 3 & \varepsilon \end{bmatrix} \quad (4.7)$$

Menyelesaikan sistem  $x(k + 1) = A_w \otimes x(k)$  dari persamaan (4.7) ketika  $x(0) = \mathbf{0}$  dengan menggunakan bantuan matlab dengan semua waktu kejadian diinisialisasi ke nol dan 8 iterasi yang dapat dilihat pada Tabel 4.1, maka diperoleh:

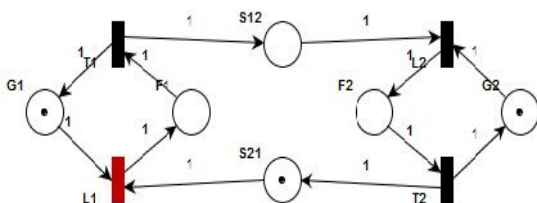
Tabel 4.1. Iterasi hasil pemodelan sistem menggunakan aljabar max plus

| Iterasi | $t_1$ | $l_1$ | $t_2$ | $l_2$ | $t_3$ | $l_3$ | $t_4$ | $l_4$ |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0       | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 1       | 7     | 6     | 5     | 4     | 6     | 5     | 4     | 3     |
| 2       | 11    | 10    | 9     | 8     | 10    | 9     | 8     | 7     |
| 3       | 15    | 14    | 13    | 12    | 14    | 13    | 12    | 11    |
| 4       | 19    | 18    | 17    | 16    | 18    | 17    | 16    | 15    |
| 5       | 23    | 22    | 21    | 20    | 22    | 21    | 20    | 19    |
| 6       | 27    | 26    | 25    | 24    | 26    | 25    | 24    | 23    |
| 7       | 31    | 30    | 29    | 28    | 30    | 29    | 28    | 27    |
| 8       | 35    | 34    | 33    | 32    | 34    | 33    | 32    | 31    |
| :       | :     | :     | :     | :     | :     | :     | :     | :     |

Pada iterasi pertama, terlihat bahwa pada detik ke-3 terjadi *lift off* kaki 'i' kemudian pada detik ke-4 kaki 'i' mengalami *touchdown*, lalu detik ke-5 kaki 'ii' *lift off* yang kemudian *touchdown* pada detik ke-6. Begitu seterusnya sampai iterasi ke-5 dengan masing-masing kejadian memiliki senggang waktu 1 detik sesuai dengan nilai  $\tau_p$  yang diberikan.

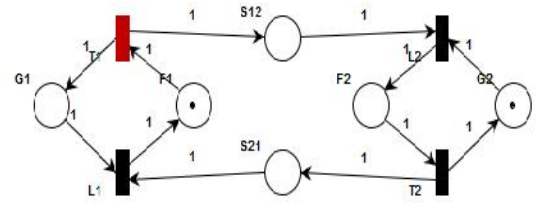
Agar lebih mudah dalam membayangkan proses gerakan pada cara berjalan *walking*, berikut ini simulasi kejadian pada Petri net cara berjalan *walking* dengan menggunakan *software PIPE* [3]:

- Gerakan pertama pada Gambar 4.1 adalah ketika kaki 'i' siap melakukan *lift off*, sehingga pada Petri net transisi  $L_1$  siap untuk di-*fire*.



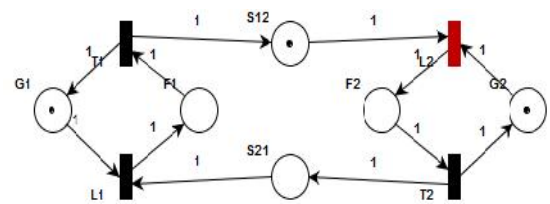
Gambar 4.1. kaki i siap untuk melakukan *lift off*.

- Setelah  $L_1$  di-*fire*, maka yang terjadi adalah kaki 'i' berada di udara atau dalam kondisi *flight* sehingga pada Petri net terlihat bahwa pada Gambar 4.2 token berpindah ke  $place F_1$  dan transisi  $T_1$  siap untuk di-*fire* atau kaki 'i' siap untuk melakukan *touchdown*.



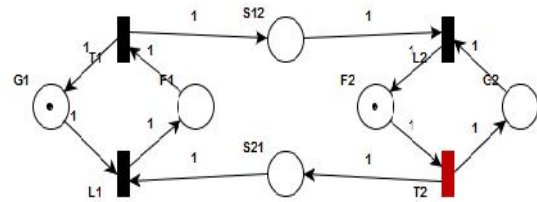
Gambar 4.2. token berpindah ke  $place F_1$  dan transisi  $T_1$  siap untuk di-*fire* atau kaki 'i' siap untuk melakukan *touchdown*.

- Setelah  $T_1$  di-*fire*, selanjutnya kaki 'ii' berada di tanah sehingga sekarang giliran kaki 'ii' yang siap untuk melakukan *lift off*. Terlihat bahwa pada Gambar 4.3 token berpindah ke  $place S_{12}$  untuk menunjukkan bahwa kaki 'ii' hanya boleh *lift off* setelah kaki 'i' telah menyentuh tanah yang mana sudah terpenuhi karena satu token sudah berada di  $G_1$ .



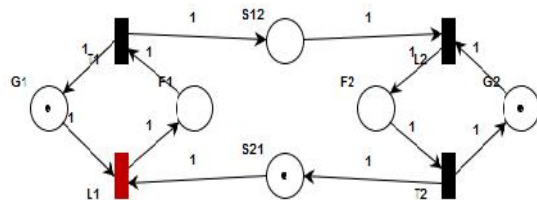
Gambar 4.3. bahwa token berpindah ke  $place S_{12}$ .

- Setelah  $L_2$  di-*fire*, maka kaki 'ii' berada di udara yang terlihat dari berpindahnya token ke  $F_2$ . Transisi berikutnya yang siap di-*fire* pada Gambar 4.4 adalah  $T_2$  karena kaki 'ii' siap melakukan *touchdown*.



Gambar 4.4. kaki 'ii' berada di udara yang terlihat dari berpindahnya token ke  $F_2$ .

- Setelah  $T_2$  di-*fire*, maka kondisi robot pada Gambar 4.5 kembali ke gerakan pertama yakni kaki 'i' siap untuk melakukan *lift off* karena kaki 'ii' telah menyentuh tanah.



Gambar 4.5. kondisi robot kembali ke gerakan pertama yakni kaki 'i' siap untuk melakukan *lift off*.

Seperti yang telah disebutkan sebelumnya bahwa kaki 'iii' memiliki waktu kejadian *lift off* dan *touchdown* yang sama dengan kaki 'i' sedangkan kaki 'iv' sama dengan kaki 'ii', maka Petri net di atas sudah sesuai dengan model gerak berjalan *walking* robot berkaki 4 pada Gambar 2.1.

## V. KESIMPULAN/RINGKASAN

Kesimpulan dari penelitian ini adalah gerak robot empat kaki bisa dimodelkan dengan menggunakan aljabar max plus dan memudahkan pencarian kaki mana yang akan naik dan yang akan turun. Saran untuk pengembangan ke depannya lebih dikembangkan lagi terhadap model gerak robot dengan kaki lebih dari empat.

## DAFTAR PUSTAKA

- [1] G.D. Lopes, et al, "Switching max-plus models for legged locomotion", IEEE International Conference on Robotics and Biomimetics. Guilin: China (2009) 221–226.
- [2] M.S. Erden and K. Leblebicioglu, "Free gait generation with reinforcement learning for a six-legged robot," *Robotics and Autonomous Systems*, Vol. 56 (2008) 199-212.
- [3] P. Holmes, et al, "The dynamics of legged locomotion: Models, analyses, and challenges". *SIAM review* Vol. 48 No. 2 (2006) 207-304.
- [4] Subiono, "Aljabar Maxplus dan Terapannya", Jurusan Maatematika Fakultas Matematika dan Ilmu Pengetahuan Alam, ITS: Surabaya (2012) version 1.1.0.
- [5] T. van den Boom and B. de Schutter, "Modelling and control of event discrete systems using switching max-plus-linear systems," *Control Engineering Practice*. Vol.14 No.10 (2006) 1199-1211.