RANGE : Jurnal Pendidikan Matematika
Dikelolah Oleh Program Studi Pendidikan Matematika Universitas Timor
ISSN Online : 2685-2373
Volume 4, Nomor 2, Januari 2023, pp. 278-290

Analisis *Coverability Tree* Petri Net Sistem Antrian Loket Pendaftaran Rumah Sakit X Di Kota Kupang

Farly Oktriany Haning[1]*, Maria Lobo[2], Suci Rahmawati[3], Elisabeth Brielin Sinu[4]
Prodi Matematika, Fakultas Sains dan Teknik, Universitas Nusa Cendana[1, 2, 4)]
Prodi Sistem Informasi, Institut Sains dan Teknologi Nahdlatul Ulama Bali[3)]
Email: farly_haning@staf.undana.ac.id

Diterima: 31 Desember 2022. Disetujui: 30 Januari 2023. Dipublikasikan: 31 Januari 2023

ABSTRAK

Model petri net dari sistem antrian di loket pendaftaran pasian pada suatu rumah sakit di Kota Kupang telah dikonstruksi. Analisis *coverability tree* dari petri net tersebut perlu dilakukan untuk melihat apakah sistem yang telah dibangun sesuai dengan kenyataan dan dapat terus berlangsung (*liveness)* atau dapat terjadi *deadlocks*. Adapun langkah-langkah penelitian ini dimulai dengan melakukan studi literatur terkait analisis petri net dan pengamatan alur layanan pada loket pendaftaran pasien. Selanjutnya, membentuk petri net dari alur layanan menggunakan bantuan software PIPE.V.4.3.2.0. Kemudian matriks *incidence A* diperoleh dengan mengurangkan entri pada matriks representasi *forward incidence* (A_f) dengan entri matriks *backward incidence* (A_b) dari petri net. Matriks *A* digunakan untuk analisis *coverability tree* dengan menggunakan vektor keadaan awal (x_0) sebagai simpul pertama dari *tree*. Simpul berikutnya dihasilkan jika terdapat transisi *enable* yang di-*fire*. Keadaan awal merupakan vektor keadaan saat belum ada transisi yang di-*fire*. Dinamika petri net yang terjadi dimodelkan dalam *coverability tree*. Hasil yang diperoleh adalah tidak terjadi *deadlocks* pada sistem karena selalu ada transisi yang *enable* atau dapat di-*fire* yaitu transisi kedatangan pasien. Pada *coverabilty tree* yang dihasilkan, keadaan setelah pendaftaran selesai dilakukan sama dengan vektor keadaan awal.

Kata kunci: antrian loket pendaftaran, *coverability tree*, matriks *incidence*, petri net.

ABSTRACT

A Petri net model of the queuing system at the patient registration counter at a hospital in Kupang City has been constructed. A coverability tree analysis of the Petri net needs to be done to see whether the system that has been built is following reality and can continue (liveness) or deadlocks can occur. This research began with a literature study on Petri net analysis and observing the flow of services at patient registration counters. Next, form a Petri net from the service flow using the PIPE.V.4.3.2.0 software. Then the incidence matrix A is obtained by subtracting the entries in the forward incidence representation matrix (A_f) from the backward incidence matrix entries (A_b) from the Petri net. Matrix A is used for tree coverability analysis by using the initial state vector (x_0) as the first node of the tree. The next node is generated if any enable transitions are fired. The initial state is the state vector when no transition has been fired. The Petri net dynamics that occur are modelled in a coverability tree. The result obtained is that there are no deadlocks in the system because there is always a transition that is enabled or can be fired, namely the transition of patient arrival. In the resulting coverability tree, the state after registration is complete is the same as the initial state vector.

Keywords*: counter registration queue, coverability tree, incidence matrix, petri net.*

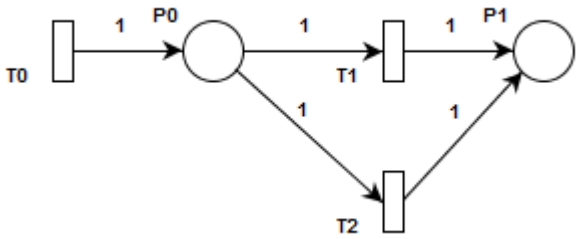
How to Cite: Haning, F. O., Lobo, M., Rahmawati, S. & Sinu, E. B. (2023). Analisis *Coverability Tree* Petri Net Sistem Antrian Loket Pendaftaran Rumah Sakit X di Kota Kupang. *Range: Jurnal Pendidikan Matematika*, 4 (2), 278-290.

Pendahuluan

Antrian pasien yang bertumpuk menjadi masalah yang harus segera ditindaklanjuti oleh pihak pemberi layanan kesehatan. Apalagi saat pandemi Covid-19 yang belum berakhir, pihak rumah sakit perlu membangun suatu layanan pendaftaran yang aman dari penyebaran penyakit, ramah dan mudah diakses. Namun, sering terjadi antrian panjang dan berdesakan di loket pendaftaran pada salah satu rumah sakit di Kota Kupang. Sistem antrian ini dapat dimodelkan menggunakan petri net karena merupakan sistem *event* diskrit (SED). Setiap perubahan keadaan pada SED bergantung pada kejadian yang telah terjadi sebelumnya(Subiono, 2009). Model petri net dapat digunakan untuk menganalisis kestabilan sistem antrian dan juga dapat dimodelkan dengan pendekatan aljabar *max plus* agar diketahui lama waktu dan keperiodikan sistem antrian (Subiono, 2015; Wattimena et al., 2012). Beberapa penelitian telah membahas penggunaan petri net dalam sistem antrian layanan kesehatan diantaranya sistem antrian poli umum di puskesmas (Munawaroh & Subiono, 2020), sistem antrian layanan pasien rawat jalan rumah sakit di Ambon (Tutupary & Lesnussa, 2013) dan rumah sakit di Yogyakarta (Osniman Paulina Maure et al., 2021), sistem antrian layanan IGD rumah sakit di Malang (Pertiwi & Khasanah, 2018), sistem antrian di Solo Peduli Klinik (Haneefa & Siswanto, 2021), sistem antrian di *medical centre* UNS Surakarta (Sulistyaningsih et al., 2020), dan sistem antrian di klinik kecantikan (Pertiwi & M Tridiana, 2020). Pada penelitian ini dibuat model petri net pada sistem antrian layanan pendaftaran pasien rawat jalan di suatu rumah sakit di Kota Kupang dan selanjutnya dianalisis *coverability tree* dari petri net tersebut. Berbeda dengan penelitian sebelumnya, pada penelitian ini membahas sistem antrian layanan kesehatan saat pandemi Covid-19.

Teori petri net dikembangkan oleh Carl Adam Petri pada tahun 1962. Petri net merupakan graf berarah yang himpunan simpulnya terbagi atas dua yaitu transisi dan *place*. Transisi digambarkan dengan persegi panjang dan *place* digambarkan dengan lingkaran. Pada petri net, transisi menyatakan kejadian yang dapat terjadi dan *place* menunjukkan keadaaan yang harus dipenuhi agar kejadian dapat terjadi (Subiono, 2009). Transisi dapat dihubungkan ke *place* oleh satu atau lebih *arc*. Namun, suatu *arc* hanya dapat menghubungkan transisi ke *place*, ataupun sebaliknya suatu *place* ke transisi. Hal ini juga mengisyaratkan bahwa petri net dapat dipandang sebagai graf berarah yang bipartit. *Place* dapat berisi token atau tanda yang merepresentasikan bahan yang berpindah dalam satu sistem Petri Net. Secara umum, petri net dengan tanda dinotasikan sebagai 4-*tuple* (P, T, A, w) dengan T menyatakan himpunan transisi pada petri net, P menyatakan himpunan *Place*, A: himpunan *arc* $A \subseteq (T \times P) \cup (P \times T)$, dan w fungsi bobot $w: A \rightarrow \{1,2,\cdots\}$.(Subiono, 2015). Pada Gambar 1 diberikan suatu contoh Petri net sederhana dengan 3 transisi dan 2 *place*. Transisi $T0$ tidak memerlukan keadaan yang harus dipenuhi

agar dapat di-*fire*. Akibatnya, transisi $T0$ selalu *enable*. Transisi $T1$ dan $T2$ membutuhkan keadaan $P0$ sebagai syarat yang harus dipenuhi agar dapat *enable*.



Gambar 1 *Contoh petri net*

Selanjutnya, diperkenalkan himpunan *place* input dan *place output* pada transisi t_j, $j \in \{1,2,\cdots,m\}$. Himpunan-himpunan ini digunakan untuk menentukan apakah suatu transisi *enable* atau tidak. Himpunan *place* input dari transisi t_j, $I(t_j)$, berisi semua *place* p_i yang dihubungkan ke transisi t_j oleh suatu *arc* (p_i, t_j) dan himpunan *place output* dari transisi t_j, $O(t_j)$, berisi semua *place* p_i yang terhubung ke transisi t_j dengan *arc* (t_j, p_i). Definisi ini dapat ditulis $I(t_j) = \{p_i | (p_i, t_j) \in A\}$ dan $O(t_j) = \{p_i | (t_j, p_i) \in A\}$.(Rudhito, 2016). Proses terjadinya perubahan pada sistem *event* diskrit yang dibangun memerlukan suatu penanda (*marking*) x yang merupakan vektor yang bernilai bilangan bulat non-negatif. Penanda lebih dikenal sebagai token dan berada pada *place*. Secara umum, x disebut vector keadaan $x = [x(p_1), x(p_2), \cdots, x(p_n)]$ dengan n adalah jumlah *place* pada petri net. Vektor keadaan menyatakan jumlah token pada setiap *place* dan digambarkan sebagai bulatan hitam (*dot*). Jika jumlah token lebih dari 5 maka cukup dituliskan dalam angka. Selanjutnya, petri net dengan keadaan awal dinotasikan sebagai 5-tuple (P, T, A, w, x_0) dengan (P, T, A, w) adalah petri net dan x_0 adalah keadaan awal (Komsiyah, 2012; Rudhito, 2016).

Dinamika petri net dapat terjadi jika semua keadaan yang diperlukan sudah terpenuhi. Keadaan yang dimaksud adalah semua *place* input p_i pada transisi t_j mempunyai jumlah token lebih dari atau sama dengan jumlah token minimum yang dibutuhkan agar suatu keadaan terpenuhi, yaitu jumlah bobot *arc* (p_i, t_j). Jika keadaan terpenuhi maka transisi dapat terjadi (*enable*). Hanya transisi yang *enable* yang dapat di-*fire* (Wattimena et al., 2012). Saat di-*fire* transisi akan mengubah keadaan sebelumnya. Secara analitik, dinamika petri net dianalisis menggunakan representasi petri net dalam matriks *incidence* (Murdianto & Santoso, 2020). Petri net dapat direpresentasikan dalam matriks yang disebut *backward incidence* (A_b) dan *forward incidence* (A_f). Kedua matriks ini masing-masing berukuran $n \times m$ dengan n adalah banyaknya *place* dan m adalah banyaknya transisi. Elemen matriks ini adalah bilangan bulat taknegatif. Elemen pada matriks *backward incidence* merupakan bobot *arc* yang menghubungkan *place*

ke transisi. Jika tidak ada *arc* yang menghubungkan *place* ke transisi maka bobot *arc* diisi nol. Sedangkan, elemen pada matriks *forward incidence* adalah bobot *arc* yang menghubungkan transisi ke *place*. Selanjutnya, matriks *incidence* (A) diperoleh dengan rumus

$$A = A_f - A_b \qquad (1)\ .$$

(Nurlela et al., 2022)

Coverability tree dibentuk dari petri net dengan keadaan awal. Mula-mula vektor keadaan awal dianggap sebagai *node root*. Selanjutnya, anak dari *node root* dihasilkan dari keadaan awal apabila salah satu transisi yang *enabled* di-*fire*. Vektor-vektor keadaan ini dihubungkan dengan sebuah sisi (*edge*) yang melambangkan transisi yang di-*fire* (Komsiyah, 2012; Sulistyaningsih et al., 2020; Wattimena et al., 2012). Dengan kata lain, transisi yang di-*fire* mengubah vektor keadaan sebelumnya. Setiap sisi pada *coverability tree* memiliki bobot sebuah transisi yang di-*fire* untuk mencapai keadaan tersebut. Dinamika ini terjadi dengan menggunakan matriks *incidence* dari petri net yang telah diperoleh sebelumnya. Proses ini menggunakan rumus

$$x(p+1) = x(p) + A\ e_j \qquad (2)$$

dengan $x(p+1)$ menyatakan vektor keadaan ke-$p+1$, A adalah matriks *incidence* dan e_j adalah kolom ke-j dari matriks identitas berukuran $m \times m$. Matriks identitas merupakan matriks persegi yang jumlah baris dan kolomnya sama dengan jumlah kolom pada matriks A dan setiap entri diagonalnya bernilai 1(Subiono, 2015).

Keadaan pada petri net yang mana tidak ada satupun transisi yang *enable* disebut sebagai *deadlock*. Transisi yang tidak *deadlock* disebut *live*. Perhatikan bahwa transisi yang *live* tidak harus *enable*. *Liveness* dapat berarti masih ada transisi yang tidak *enabled* tapi masih mungkin untuk *enable* (Wattimena et al., 2012). Jika terjadi *deadlock* maka petri net tersebut masih belum dapat digunakan untuk simulasi ataupun analisis lebih lanjut. Analisis *Coverability Tree* bertujuan untuk mengetahui apakah semua transisi akan tetap *live* atau ada kemungkinan terjadinya *deadlock*.

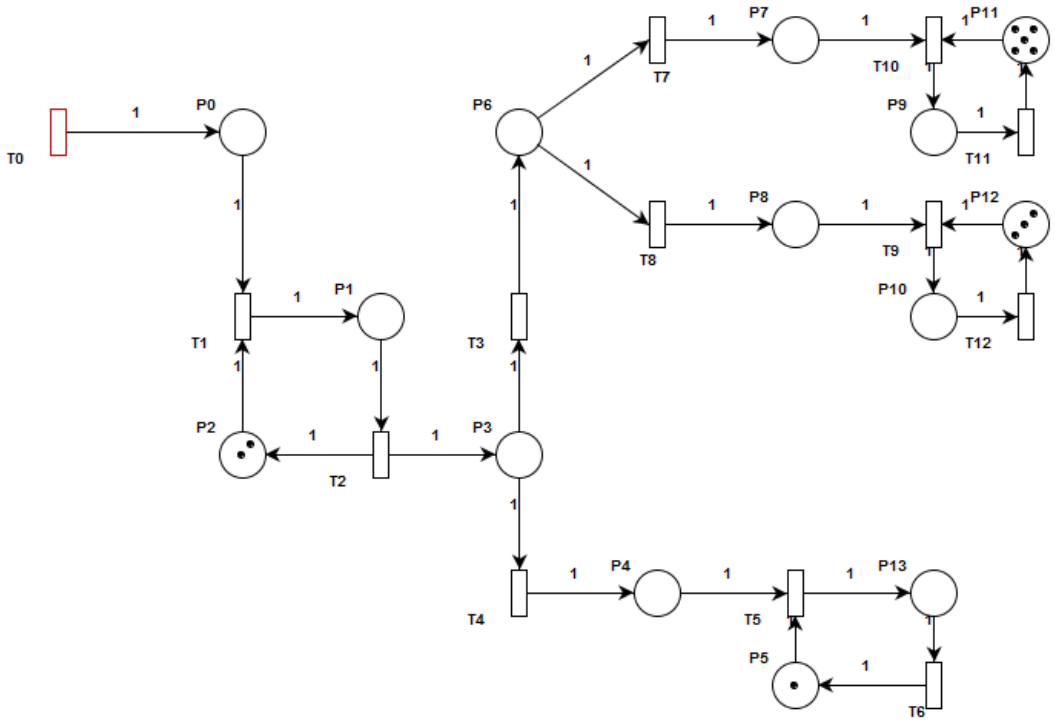
Metode Penelitian

Penelitian ini dilakukan di suatu rumah sakit di Kota Kupang pada bulan Juli 2022. Penelitian ini masuk dalam penelitian deskriptif kualitatif dengan analisisnya menggunakan bantuan perangkat lunak PIPE.V.4.3.2.0 yang dapat diunduh secara gratis. Prosedur penelitian mengikuti langkah-langkah berikut: (i) studi literatur materi terkait analisis petri net, (ii) melakukan pengamatan alur layanan pada loket pendaftaran pasien, (iii) membentuk petri net dari alur layanan menggunakan bantuan perangkat lunak PIPE.V.4.3.2.0, (iv) menghasilkan matriks *incidence* yang diperoleh dari mengurangkan setiap entri pada matriks representasi *forward incidence* dengan entri pada matriks *backward incidence*

menggunakan rumus pada Persamaan (1), (v) menetapkan vektor keadaan awal sebagai simpul pertama dari *coverability tree*, (vi) menggambarkan *coverability tree* dengan simpul pertama adalah vektor keadaan awal, dan simpul berikutnya adalah vektor keadaan setelah pem-*fire*-an suatu transisi yang *enable*, dan (vii) memaparkan hasil analisis *coverability tree* dan memberi simpulan.

Hasil Penelitian dan Pembahasan

Alur layanan pendaftaran pasien rawat jalan pada suatu rumah sakit di Kota Kupang saat pandemi Covid-19 mengikuti protokol yang telah ditetapkan oleh kementrian kesehatan (Kemenkes, 2021).
Setiap pasien/ keluarga pasien yang akan melakukan pendaftaran perlu melakukan skrining terlebih dahulu. Jika ditemui ada gejala atau ciri-ciri seperti covid 19 maka pemeriksaan dilanjutkan dengan prosedur penanganan pasien covid. Jika tidak ada gejala maka pasien dapat melanjutkan proses pendaftaran di loket ataupun mesin anjungan pendaftaran mandiri (APM). Mesin APM digunakan apabila pasien merupakan pasien lama yang akan berobat. Sebaliknya, jika pasien adalah pasien yang baru sehingga belum memiliki nomor rekam medik (RM) maka harus mendaftar melalui loket pendaftaran. Setelah pendaftaran pasien dapat melanjutkan ke poliklinik yang dituju ataupun dapat kembali ke rumah. Petri net dari alur layanan pendaftaran pasien terlihat pada Gambar 2.



Gambar 2. *Petri net dari alur layanan pendaftaran pasien rawat jalan*

Petri net pada Gambar 2 memiliki 13 transisi, yaitu t_0: Kedatangan pasien di rumah sakit untuk pemeriksaan, t_1: Pasien mulai di-skrining, t_2: Pasien selesai di-skrining, t_3: Pasien menuju loket pendaftaran, t_4: Pasien mulai antri di Triase Covid 19, t_5: Pasien dilayani dokter di Triase Covid 19, t_6: Pasien selesai dilayani di Triase Covid 19, t_7: Pasien baru mulai mengantri di loket 2-6, t_8: Pasien lama mulai mengantri di mesin anjungan pendaftaran mandiri, t_9: Pasien lama mendaftar di mesin anjungan pendaftaran mandiri, t_{10}: Pasien baru mulai mendaftar di loket 2-6, t_{11}: Pasien baru telah selesai melakukan pendaftaran di loket 2-6, dan t_{12}: Pasien lama selesai mendaftar di mesin anjungan pendaftaran mandiri.

Pada petri net di Gambar 2 terdapat 14 *place*, yaitu p_0: Pasien pergi ke antrian skrining, p_1: Pasien sedang diskrining oleh petugas, p_2: Petugas skrining idle atau sedang tidak sibuk, p_3: Pasien selesai di-skrining, p_4: Pasien gejala covid mulai mengantri di Triase Covid 19 untuk pemeriksaan lanjut, p_5: Dokter di Triase Covid 19 sedang idle, p_6: Pasien non covid pergi ke loket pendaftaran, p_7: Pasien non covid mulai mengantri di loket pendaftaran, p_8: Pasien non covid mulai mengantri di mesin pendaftaran mandiri, p_9: Pasien non covid dilayani petugas loket pendaftaran, p_{10}: Pasien non covid mendaftar di mesin pendaftaran mandiri, p_{11}: Petugas loket sedang idle, p_{12}: Mesin anjungan pendaftaran mandiri sedang idle, p_{13}: Pasien covid sedang dilayani dokter.

Selanjutnya dikaji representasi petri net dalam matriks yaitu matriks *backward incidence* dan matriks *forward incidence*. Matriks representasi untuk petri net pada Gambar 2 memiliki ordo 14×13. Elemen pada matriks *forward incidence* adalah bobot pada *arc* (t_j, p_i) dan elemen pada matriks *backward incidence* adalah bobot pada *arc* (p_i, t_j).

$$A_f = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0 \end{bmatrix} \text{ dan } A_b = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0 \end{bmatrix}$$

Matriks *incidence* diperoleh dari rumus $A = A_f - A_b$

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Pada mulanya hanya transisi t_0 yang *enables* yaitu transisi kedatangan pasien di rumah sakit untuk pemeriksaan. Transisi ini dapat terus di-*fire* dengan asumsi loket sedang dibuka untuk umum. Jika ada pasien yang datang, maka keadaan awal pada sistem akan berubah. Jika keadaan berubah maka dapat dilihat transisi lain yang *enable* untuk di-*fire*. Proses ini dikenal sebagai dinamika petri net.

Untuk mengamati dinamika petri net diperlukan keadaan awal petri net pada **Error! Reference source not found.** yaitu

$$x_0 = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 0 \; 0 \quad 0 \quad 0 \quad 0 \quad 5 \; 3 \quad 0 \;]^T.$$

Pada keadaan ini, terdapat 2 token pada *place* p_2, yang berarti terdapat dua petugas skrining. Pada *place* p_5 terdapat 1 dokter di Triase Covid-19. *Place* p_{11} menandakan banyaknya loket pendaftaran yang dibuka dan *place* p_{12} menyatakan banyaknya mesin anjungan pendaftaran mandiri yang tersedia untuk digunakan. Pada keadaan ini, satu-satunya transisi yang dapat di-*fire* adalah transisi t_0.

Selanjutnya dengan menggunakan rumus pada Persamaan (2) diperoleh keadaan setelah transisi t_0 di-*fire* yaitu $x(1) = x(0) + A\, e_1$

$$x(1) = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \\ 3 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \\ 3 \\ 0 \end{bmatrix}$$

Keadaan $x(1)$ menandakan sudah terdapat token di *place* p_0 mengakibatkan transisi t_1 menjadi *enable*. Dalam penerapannya, keadaan ini menandakan sudah ada pasien rawat jalan yang datang ke rumah sakit sehingga pasien sudah mulai masuk ke antrian untuk proses skrining. Perhatikan bahwa t_0 akan selalu *enable* berarti pasien yang datang tidak dibatasi jumlahnya. Apabila t_0 selalu di-*fire* maka token di *place* p_1 akan terus bertambah. Untuk itu, hanya ditinjau proses saat t_1 di-*fire*.

Keadaan selanjutnya adalah akibat dari t_1 di-*fire*.

$$x(2) = x(1) + A\,e_2 = [0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 5 \quad 3 \quad 0 \quad]^T$$

Keadaan $x(2)$ menunjukkan bahwa petugas sedang melakukan skrining pada pasien yang telah datang. Terlihat jumlah token pada *place* p_0 telah kembali menjadi 0 dan jumlah token pada *place* p_2 yang semula berjumlah 2 telah berkurang menjadi 1. Sementara pada *place* p_1 telah berisi 1 token yang menandakan pasien sedang di-skrining. Transisi yang *enable* adalah t_0 dan t_2. Oleh karena t_0 akan mengulang proses kedatangan pasien maka pada pembahasan ini dipilih transisi t_2 untuk di-*fire* dan keadaan yang dihasilkan adalah $x(3)$.

$$x(3) = x(2) + A\,e_3 = [0 \quad 0 \quad 2 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 5 \quad 3 \quad 0 \quad]^T.$$

Keadaan $x(3)$ menyatakan bahwa pada *place* p_2 petugas skrining kembali idle dan pasien telah selesai diperiksa covid-19 sehingga token berpindah ke *place* p_4. Transisi yang *enable* adalah transisi t_3 dan t_4.

Jika transisi t_3 yang di-*fire* akan diperoleh keadaaan $x(4)$

$$x(4) = x(3) + A\,e_4 = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 5 \quad 3 \quad 0 \quad]^T.$$

Keadaan $x(4)$ menyatakan bahwa token pada *place* p_3 telah berpindah ke p_6. Jika transisi t_3 yang terjadi maka pasien telah selesai skrining dan tidak bergejala covid sehingga melanjutkan ke loket pendaftaran (p_6). Transisi yang *enable* adalah transisi t_7 dan t_8. Pem-*fire*-an transisi t_7 atau t_8 akan dibahas setelah kasus t_4 yang di-*fire* terlebih dahulu.

Jika transisi t_4 yang di-*fire* akan diperoleh keadaaan $x(5)$

$$x(5) = x(3) + A\,e_5 = [0 \quad 0 \quad 2 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 5 \quad 3 \quad 0 \quad]^T.$$

Keadaan $x(5)$ menunjukkan bahwa token pada *place* p_3 berpindah ke p_4. Artinya pasien telah selesai di-skrining covid-19 dan bergejala covid 19 sehingga pasien masuk ke Triase Covid 19 untuk penanganan lebih lanjut oleh dokter. Transisi t_5 menjadi *enable*.

Apabila transisi t_5 di-*fire* maka keadaaan yang terjadi $x(6)$

$$x(6) = x(5) + A\,e_6 = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 5 \quad 3 \quad 1 \quad]^T$$

Keadaan $x(6)$ adalah kejadian token pada *place* p_4 dan p_5 berpindah ke p_{13}. Artinya pasien covid 19 sedang dilayani oleh dokter ditandai dengan token pada p_4 menjadi kosong. Token pada p_5 juga kosong artinya dokter sedang sibuk. Apabila ada pasien covid 19 berikutnya maka ia harus menunggu waktu

antrian. Selanjutnya, transisi t_6 menjadi *enable* menandakan bahwa pasien telah selesai ditangani oleh dokter. Pem-*fire*-an transisi t_6 menghasilkan keadaan $x(7)$

$$x(7) = x(6) + A\, e_7 = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 0 \; 0 \quad 0 \quad 0 \quad 0 \quad 5 \; 3 \quad 0 \;]^T$$

Keadaan $x(7)$ menandakan token di p_{13} telah kosong karena pasien telah selesai dilayani dan dokter di p_5 kembali idle untuk melayani pasien berikutnya. Pada keadaan ini, satu-satunya transisi yang *enable* adalah t_0 sehingga keadaan akan kembali seperti keadaan awal x_0.

Selanjutnya dibahas keadaan yang dihasilkan saat t_3 di-*fire* yaitu t_7 atau t_8 yang *enable*. Pembahasan dilakukan telebih dahulu untuk keadaan saat transisi t_7 yang di-*fire*. Jika t_7 yang di-*fire* maka keadaan menjad i$x(8)$

$$x(8) = x(4) + A\, e_8 = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 0 \; 1 \quad 0 \quad 0 \quad 0 \quad 5 \; 3 \quad 0 \;]^T$$

Keadaan $x(8)$ menunjukkan bahwa token pada p_6 berpindah ke p_7. Artinya pasien melanjutkan pendaftaran di loket pendaftaran. Transisi yang *enable* adalah t_{10}. Apabila transisi t_{10} di-*fire* maka keadaan berubah menjadi $x(10)$

$$x(10) = x(8) + A\, e_{10} = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 0 \; 0 \quad 0 \quad 1 \quad 0 \quad 4 \; 3 \quad 0 \;]^T$$

Keadaan $x(10)$ menandakan bahwa pasien sedang didaftarkan oleh petugas di loket pendaftaran. Ini berakibat, transisi yang *enable* adalah transisi t_{11}. Apabila t_{11} di-fire keadaan akan menjadi$x(11)$

$$x(11) = x(10) + A\, e_{11} = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 0 \; 0 \quad 0 \quad 0 \quad 0 \quad 5 \; 3 \quad 0 \;]^T$$

Keadaaan $x(11)$ sama dengan keadaan awal x_0 karena pasien telah selesai dilayani dan meninggalkan loket pendaftaran.

Pembahasan dilanjutkan untuk keadaan saat transisi t_8 yang di-*fire*. Jika transisi t_8 di-*fire* maka keadaan pada $x(4)$ akan menjadi $x(9)$

$$x(9) = x(4) + A\, e_8 = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 0 \; 0 \quad 1 \quad 0 \quad 0 \quad 5 \; 3 \quad 0 \;]^T$$

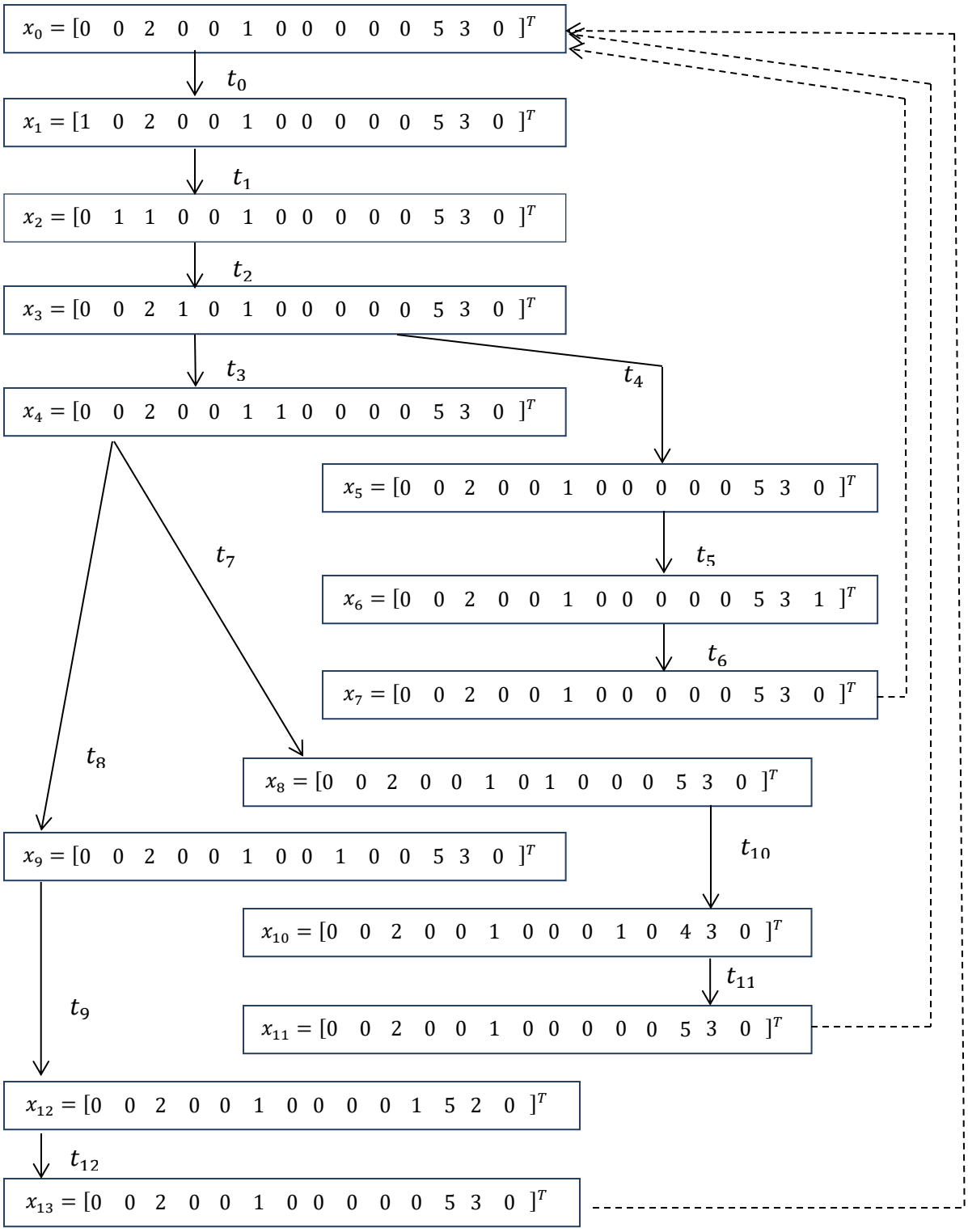
Keadaan $x(9)$ menunjukkan bahwa pasien ingin melakukan pendaftaran secara mandiri menggunakan mesin anjungan pendaftaran mandiri (APM). Keadaaan ini mengimplikasikan sudah ada pasien yang mengantri untuk menggunakan mesin APM sehingga transisi yang *enable* adalah transisi t_9. Jika transisi t_9 di-*fire* akan menghasilkan keadaan$x(12)$

$$x(12) = x(9) + A\, e_9 = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 0 \; 0 \quad 0 \quad 0 \quad 1 \quad 5 \; 2 \quad 0 \;]^T$$

Pada keadaan ini, token di *place* p_9 berpindah ke p_{10} dan token di p_{12} berkurang 1. Ini berarti, pasien sedang melakukan pendaftaran melalui mesin APM sehingga mesin APM yang idle tersisa 2. Transisi yang *enable* adalah transisi t_{12}. Pem-*fire*-an transisi t_{12} akan menghasilkan keadaan $x(13)$

$$x(13) = x(12) + A\, e_{13} = [0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1 \quad 0 \; 0 \quad 0 \quad 0 \quad 0 \quad 5 \; 3 \quad 0 \;]^T$$

Keadaan $x(13)$ telah sama dengan keadaan x_0 sebab pasien telah selesai mendaftar di APM dan meninggalkan tempat pendaftaran. Dinamika petri net ini digambarkan dalam *coverability tree* yang tersaji pada Gambar 3.



Gambar 3. *Coverability tree dari petri net alur layanan pendaftaran pasien*

Pada Gambar 3 terlihat keadaan $x(0), x(7), x(11)$ dan $x(13)$ memiliki *coverability* yang sama ditandai dengan garis putus-putus. *Node* pertama dari *coverability tree* adalah *root node* merupakan keadaan awal petri net $x(0)$. Keadaan selanjutnya disebut sebagai anak (*child)* dari *root node*. Keadaan $x(1)$ merupakan anak dari $x(0)$ dan keadaan $x(2)$ merupakan anak dari $x(1)$. Demikian juga, anak dari $x(2)$ adalah $x(3)$. Selanjutnya, keadaan $x(3)$ memiliki dua anak yaitu $x(4)$ dan $x(5)$. Sementara, $x(4)$ juga memiliki dua anak yaitu $x(8)$ dan $x(9)$.

Pada Gambar 3, terlihat tidak terjadi *deadlock* pada petri net. Hal ini dikarenakan setiap *node* keadaan selalu mempunyai anak dan transisi lanjutan yang dapat di-*fire*. Setelah keadaan $x(7)$ terjadi transisi yang dapat di-*fire* adalah transisi t_0. Demikian juga keadaan $x(11)$ dan $x(13)$ akan memiliki t_0 sebagai transisi yang di-*fire*. Ini berarti, selalu ada transisi yang *enable* sehingga tidak terjadi *deadlock*. Dengan kata lain, semua transisi pada petri net selalu *live*, sebab selalu ada kemungkinan untuk *enable*. Keadaan ini menjamin sistem akan dapat terus berlangsung dan juga sesuai dengan keadaan sebenarnya. Walaupun petri net ini tidak memenuhi sifat *boundness* karena jumlah token pada beberapa *place* petugas layanan lebih dari satu (Komsiyah, 2012), tetapi Petri net masih dapat digunakan untuk analisis lebih lanjut. Petri net ini dapat dimodelkan menggunakan aljabar *max plus* seperti pada (Hardiyanti et al., 2017; Osniman P Maure & Rudhito, 2019; Osniman Paulina Maure et al., 2021; Tutupary & Lesnussa, 2013). Petri net ini dapat dilanjutkan untuk analisis kestabilan sistem antrian seperti pada (Sya'diyah, 2021) dan analisis keperiodikannya dengan melihat polinomial karakteristik dari matriks *incidence* (Subiono, 2015).

Kesimpulan

Telah diperoleh *coverability tree* dari petri net yang telah dibangun pada sistem antrian loket pendaftaran pasien rawat jalan di suatu rumah sakit di Kota Kupang. *Coverability tree* dibentuk dari petri net dengan keadaan awal (P, T, A, w, x_0). Vektor keadaan awal x_0 digambar sebagai *node root*. Apabila salah satu transisi yang *enable* di-*fire* maka menghasilkan vektor keadaan yang dinamakan anak dari *node root*. Setiap transisi yang di-*fire* dilambangkan dengan sisi yang menghubungkan vektor-vektor keadaan yang dihasilkan. Hasil analisis menunjukkan bahwa tidak terjadi *deadlock* sebab semua transisi pada petri net selalu *live*. Namun, petri net tersebut belum memenuhi *boundedness* (*safe*) sebab terdapat beberapa *place* yang tokennya lebih dari satu.

Beberapa kajian dapat dilanjutkan dari penelitian ini adalah analisis kestabilan petri net sistem antrian, analisis model aljabar *max plus* dari petri net berwaktu, ataupun analisis keperiodikan dari model aljabar *max plus* yang dibangun.

Ucapan Terima Kasih

Penelitian ini dapat terlaksana dengan dukungan dana penelitian dengan DIPA UNDANA 2022. Ucapan terimakasih untuk prodi matematika FST Undana yang telah memberikan kesempatan untuk melakukan penelitian hingga terpublikasinya hasil penelitian ini.

Daftar Pustaka

Haneefa, G. P. T., & Siswanto. (2021). Petri Net Model and Max-Plus Algebra in Outpatient Care at Solo Peduli Clinic, Surakarta. *Journal of Physics: Conference Series*, *1776*(1). https://doi.org/10.1088/1742-6596/1776/1/012047

Hardiyanti, S. A., Yuniwati, I., Divi Yustita, A., & Banyuwangi, P. N. (2017). Bentuk Petri Net Dan Model Aljabar Max Plus Pada Sistem Pelayanan Pasien Rawat Jalan Rumah Sakit Al Huda Genteng, Banyuwangi. *Jurnal UJMC*, *3*(2), 1–8.

Kemenkes. (2021). *Pedoman pelayanan Rumah Sakit Pada Masa Pandemi Covid-19* (Revisi 1). Kementerian Kesehatan RI. https://www.kemkes.go.id/downloads/resources/download/info-terkini/COVID-19/Pedoman-Pelayanan-Rumah-Sakit-Pada-Masa-Pandemi-COVID-19_edisi-revisi-1.pdf

Komsiyah, S. (2012). *Model Petri Net Tak Berwaktu Pada Sistem Produksi (Batch Plant) Dan Simulasinya Dengan Pipe2*, *12*(9), 152–164.

Maure, Osniman P, & Rudhito, M. A. (2019). Model Aljabar Max - Plus Pada Sistem Antrian Pelayanan Penerbitan Surat Izin Usaha Perdagangan Bahan Berbahaya Max-Plus Algebra Model On Service Queue System Publishing Of Trade Materials License Hazardous Substances. *Asimtot: Jurnal Kependidikan Matematika*, *139*(2), 139–146.

Maure, Osniman Paulina, Ningsi, G. P., & Nay, F. A. (2021). Pemodelan Sistem Antrian Pasien Rawat Jalan Menggunakan Petri Net Dan Aljabar Max-Plus: Studi Kasus Rsu Di Yogyakarta. *Jurnal Matematika*, *1*(1), 21–35.

Munawaroh, M., & Subiono. (2020). Petri Net Dan Model Aljabar Max Plus Pada Sistem Pelayanan Pasien Poli Umum Di Puskesmas XYZ. *SNasPPM: Prosiding Seminar Nasional Penelitian Dan Pengabdian Masyarakat*, *5*(1), 123–131. http://prosiding.unirow.ac.id/index.php/SNasPPM/article/view/323

Murdianto, D., & Santoso, H. (2020). Pemodelan Prosedur Karantina Pendatang Dalam Rangka Pencegahan Covid-19 Di Kota Tarakan Menggunakan Petri Net. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, *14*(4), 587–596. https://doi.org/10.30598/barekengvol14iss4pp587-596

Nurlela, N., Faisol, A., & Fitriani, F. (2022). Model Petri Net Sistem Pembayaran Pajak Kendaraan Bermotor Jenis 5 Tahun. *Jambura Journal of Mathematics*, *4*(1), 33–40. https://doi.org/10.34312/jjom.v4i1.11158

Pertiwi, R. I., & Khasanah, F. (2018). Aplikasi Petri Net Pada Sistem Pelayanan Igd Rsud Dr. Saiful Anwar Malang. *Science Tech: Jurnal Ilmu Pengetahuan Dan Teknologi*, *4*(2), 75–79. https://doi.org/10.30738/jst.v4i2.2763

Pertiwi, R. I., & M Tridiana, L. (2020). Model Petri Net Dari Antrian Klinik Kecantikan Serta Aplikasinya Pada Aljabar Maxplus. *MAp (Mathematics and Application) Journal*, *2*(1), 34–40.

Rudhito, M. A. (2016). Aljabar max-plus dan penerapannya. In *Universitas Sanata Dharma Yogyakarta*.

Subiono. (2015). *Aljabar Min-Max Plus dan Terapannya*.

Subiono, S. (2009). Aljabar Maxplus dan Aplikasinya : Model Sistem Antrian. *Limits: Journal of Mathematics and Its Applications*, *6*(1), 49. https://doi.org/10.12962/j1829605x.v6i1.1431

Sulistyaningsih, T., Siswanto, & Pangadi. (2020). Petri Net Model and Max-Plus Algebra on Queue in Clinic UNS Medical Center. *Journal of Physics: Conference Series*, *1494*(1).

https://doi.org/10.1088/1742-6596/1494/1/012004

Sya'diyah, Z. (2021). Kestabilan Model Petri Net Dari Sistem Pembayaran Tagihan Listrik PT. PLN (Persero) Rayon Ambon Timur. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, *15*(4), 601–606. https://doi.org/10.30598/barekengvol15iss4pp601-606

Tutupary, F. S., & Lesnussa, Y. A. (2013). Aplikasi Petri Net Pada Sistem Pelayanan Pasien Rawat Jalan Peserta Askes Di Rumah Sakit Umum Daerah DR. Haulussy Ambon. *Gamatika*, *III*(2), 147–154.

Wattimena, F. N., Pentury, T., & Lesnussa, Y. A. (2012). Aplikasi Petri Net Pada Sistem Pembayaran Tagihan Listrik PT. PLN (Persero) Rayon Ambon Timur. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, *6*(1), 23–30. https://doi.org/10.30598/barekengvol6iss1pp23-30