



TESIS - SM 142501

STRUKTUR HIRARKIS JALUR KERETA API SEMI- DOUBLE TRACK MENGGUNAKAN PETRI NET DAN ALJABAR MAX-PLUS

TRI UTOMO
NRP 1213 201 015

DOSEN PEMBIMBING
Dr. Subiono, M.S.

PROGRAM MAGISTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015



TESIS - SM 142501

STRUKTUR HIRARKIS JALUR KERETA API SEMI- DOUBLE TRACK MENGGUNAKAN PETRI NET DAN ALJABAR MAX-PLUS

TRI UTOMO
NRP 1213 201 015

DOSEN PEMBIMBING
Dr. Subiono, M.S.

PROGRAM MAGISTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015



THESIS - SM 142501

HIERARCHICAL STRUCTURE FOR SEMI-DOUBLE TRACK OF RAILWAY LINE USING PETRI NETS AND MAX-PLUS ALGEBRA

TRI UTOMO
NRP 1213 201 015

SUPERVISOR
Dr. Subiono, M.S.

MAGISTER'S DEGREE
MATHEMATICS DEPARTMENT
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2015

**STRUKTUR HIRARKIS JALUR KERETA API SEMI-DOUBLE
TRACK MENGGUNAKAN PETRI NET DAN ALJABAR MAX-
PLUS**

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sains (M.Si.)

di
Institut Teknologi Sepuluh Nopember

Oleh:
TRI UTOMO
NRP. 1213 201 015

Tanggal Ujian : 13 Mei 2015
Periode Wisuda : September 2015

Disetujui oleh:

Dr. Subiono, M.S.
NIP. 19570411 198403 1 001

(Pembimbing)

Dr. Hariyanto, M.Si.
NIP. 19530414 198203 1 002

(Penguji)

Dr. Mardijah, M.T.
NIP. 19670114 199102 2 001

(Penguji)

Direktur Program Pascasarjana,

Prof. Dr. Ir. Adi Soeprijanto, M.T.
NIP. 19640405 199002 1 001

STRUKTUR HIRARKIS JALUR KERETA API SEMI-DOUBLE TRACK MENGGUNAKAN PETRI NET DAN ALJABAR MAX-PLUS

Nama Mahasiswa	: Tri Utomo
NRP	: 1213 201 015
Dosen Pembimbing	: Dr. Subiono, M.S.

ABSTRAK

Masih tingginya perbedaan pembangunan antara kota-kota besar dan daerah di sekitarnya menyebabkan terjadinya proses perpindahan yaitu berupa pengangkutan barang maupun manusianya itu sendiri (arus urbanisasi) atau biasa dikenal dengan istilah transportasi. Dalam mengatasi permasalahan transportasi ini pemerintah menetapkan beberapa kebijakan salah satunya yaitu pembangunan jalur kereta api *double track* yang saat ini untuk Lintas Utara Pulau Jawa telah selesai pengerjaannya dan untuk Lintas Selatan Pulau Jawa masih terhambat dengan masalah pembebasan lahan. Penelitian ini dimaksudkan untuk mencari solusi alternatif dalam mengatasi permasalahan transportasi khususnya jalur kereta api dengan mengoptimalkan penggunaan jalur kereta api *single track* yang dalam penelitian ini diberi istilah *semi-double track* menggunakan Petri Net dan Aljabar Max-Plus. Dalam mengerjakan penelitian ini dikonstruksi model Petri Net dari jalur kereta api *semi-double track*. Model tersebut dikonstruksi berdasarkan aturan struktur hirarkis arah jalur lintasannya, sehingga tidak terjadi *deadlock*. Kemudian mengkonstruksi model Aljabar Max-Plus dari model Petri Net yang sudah didapatkan dan terakhir menganalisis sifat keperiodikan sistem tersebut berdasarkan model Aljabar Max-Plus yaitu didapatkan nilai eigen $\lambda = 32$ yang berarti bahwa keberangkatan kereta api yang berjalan searah pada tiap-tiap stasiun adalah setiap 32 menit sekali.

Kata kunci : Aljabar Max-Plus, Petri Net, Railways, Semi-Double Track, Struktur Hirarkis.

HIERARCHICAL STRUCTURE FOR SEMI-DOUBLE TRACK OF RAILWAY LINE USING PETRI NETS AND MAX-PLUS ALGEBRA

Name	: Tri Utomo
NRP	: 1213 201 015
Supervisor	: Dr. Subiono, M.S.

ABSTRACT

The persistently inequality of development between major cities and surrounding areas caused the displacement process, that are transport of supplies or human itself (urbanization) or commonly known as the transportation. The government has set a number of policies to overcome transportation problems, for example construct a double track railway line which is now across the Northern of Java has completed and for across the Southern of Java still hampered by land acquisition problems. This study aimed to find an alternative solution to overcome the problems of transportation, especially railway line by optimizing of using single track railway line which is given the term semi-double track using Petri Nets and Max-Plus Algebra. In this paper we made a model of Petri Nets of railway line semi-double track. The Petri Nets model is built based on the priority rules, so the Petri Net model is not deadlock. Then we made a model of Max-Plus Algebra based on the Petri Nets model and the last research is analyze the properties of the system periodicity depend on Max-Plus Algebra model and obtained eigenvalue $\lambda = 32$ which means that the departure of the trains that run in the same direction at each station is every 32 minutes.

Keywords : *Hieracichal Structure, Max-Plus Algebra, Petri Nets, Railways, Semi-Double Track.*

KATA PENGANTAR



Alhamdulillahirobbil 'alamin, segala puji syukur ke hadirat Alloh SWT atas limpahan rahmat, taufiq dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tesis yang berjudul “Struktur Hirarkis Jalur Kereta Api Semi-Double Track Menggunakan Petri Net dan Aljabar Max-Plus” ini dengan baik. Sholawat serta salam semoga senantiasa tercurahkan kepada Nabi besar Muhammad SAW sebagai *Uswatun Khasanah* dalam meraih kesuksesan di dunia dan akhirat.

Selanjutnya penulis haturkan ucapan terima kasih seiring do’a dan harapan *jazakumullohu akhsanal jaza’* kepada semua pihak yang telah membantu penulis dalam menyelesaikan Tesis ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Prof. Ir. Joni Hermana, M.Sc.E.S., Ph.D. selaku Rektor Institut Teknologi Sepuluh Nopember Surabaya.
2. Prof. Dr. Ir. Adi Soeprijanto, M.T., selaku Direktur Program Pascasarjana Institut Teknologi Sepuluh Nopember Surabaya.
3. Prof. Dr. Dra. Erna Apriliani, M.Si., selaku Ketua Jurusan Matematika Institut Teknologi Sepuluh Nopember Surabaya.
4. Prof. Dr. Basuki Widodo, M.Sc., selaku Dosen Wali yang telah memberikan motivasi, arahan, dan bimbingannya.
5. Dr. Subiono, M.Sc., selaku Koordinator Program Studi Pascasarjana Matematika dan selaku Dosen Pembimbing, yang telah memberikan pengarahan dan pengalaman yang berharga.
6. Dr. Hariyanto, M.Si. dan Dr. Mardlijah, M.T., selaku Dosen Penguji yang telah memberikan masukan dan juga saran kepada penulis sehingga Tesis ini dapat diselesaikan dengan baik.
7. Seluruh Dosen Jurusan Matematika Institut Teknologi Sepuluh Nopember Surabaya, terimakasih atas seluruh ilmu, nasihat, dan bimbingannya, serta seluruh Staff Administrasi, terima kasih atas segala bantuannya.

8. Bapak, Ibu, dan sekeluarga tercinta, yang senantiasa memberikan do'a dan restunya kepada penulis dalam menuntut ilmu.
9. Teman-teman Pascasarjana Matematika ITS angkatan 2013, terima kasih atas kenangan yang kalian berikan.
10. Teman-teman tercinta Pondok Pesantren Modern Raden Paku Trenggalek terutama angkatan III, teman-teman Wearness Education Center Malang, dan teman-teman Jurusan Matematika Universitas Islam Negeri Maulana Malik Ibrahim Malang terutama angkatan 2007, yang telah memberikan motivasi dan inspirasi kepada penulis.
11. Seluruh Staff PT Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya.

Semoga Allah SWT selalu memberikan anugerah dan karunia-Nya kepada semua pihak yang telah membantu penulis dalam menyelesaikan Tesis ini, Amin.

Penulis menyadari bahwa dalam penulisan Tesis ini masih banyak kekurangan, sehingga kritik dan saran dari pembaca sangat penulis harapkan untuk perbaikan kedepannya. Semoga Tesis ini bermanfaat dan dapat menambah wawasan keilmuan khususnya ilmu matematika dan terapannya, Amin.

Surabaya, 20 Mei 2015

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
DAFTAR SIMBOL	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	7
2.1 Kajian Pustaka	7
2.2 Petri Net	9
2.2.1 Definisi dan Notasi dalam Petri Net.....	9
2.2.2 Petri Net Bertanda dan Ruang Keadaan.....	11
2.2.3 Dinamika Petri Net.....	12
2.3 Aljabar Max-Plus	13
2.3.1 Definisi dan Notasi dalam Aljabar Max-Plus	13
2.3.2 Vektor dan Matriks atas Aljabar Max-Plus.....	15
2.3.3 Nilai Eigen dan Vektor Eigen	17
2.3.4 Solusi Persamaan Linear	17
2.4 Persamaan Relasi Rekurensi	20
BAB 3 METODA PENELITIAN.....	23
BAB 4 HASIL DAN PEMBAHASAN	25
4.1 Analisis Permasalahan	25

4.2	Model Petri Net	31
4.3	Model Aljabar Max-Plus	37
BAB 5	KESIMPULAN	57
5.1	Kesimpulan.....	57
5.2	Saran.....	58
DAFTAR PUSTAKA		59
LAMPIRAN		61
BIOGRAFI PENULIS		69

DAFTAR GAMBAR

Gambar 1.1	Ilustrasi Permasalahan dari Dua Stasiun Kereta Api (a) Dua Stasiun yang Digambarkan dalam Sebuah Graf, (b) Ilustrasi Jalur Kereta Api <i>Single Track</i> , (c) Ilustrasi Jalur Kereta Api <i>Double Track</i> , (d) Ilustrasi Jalur Kereta Api <i>Semi-Double Track</i>	3
Gambar 2.1	Penelitian Terdahulu.....	7
Gambar 2.2	Ilustrasi Alur Penyelesaian Permasalahan.....	9
Gambar 2.3	Contoh Petri Net Sederhana	10
Gambar 2.4	Keadaan Setelah t_1 di- <i>fire</i> yang Pertama (Kiri) dan Kedua (Kanan).....	13
Gambar 2.5	Contoh Keadaan <i>Deadlock</i>	13
Gambar 4.1	Peta Jalur Kereta Api Waru-Sidoarjo	26
Gambar 4.2	Kondisi Stasiun Waru, Stasiun Gedangan, dan Stasiun Sidoarjo.....	26
Gambar 4.3	Gambaran Umum Jalur Kereta Api Waru-Sidoarjo	27
Gambar 4.4	Ilustrasi Keadaan yang Harus Dihindari.....	27
Gambar 4.5	Contoh Kasus yang Memerlukan Prioritas.....	28
Gambar 4.6	Keadaan Sistem Jika Urutan Penyelesaian Tidak Ada Aturan Prioritas.....	29
Gambar 4.7	Urutan Prioritas Penyelesaian Kasus 1	30
Gambar 4.8	Kejadian <i>Bottleneck</i> Lainnya pada Kasus 1	30
Gambar 4.9	Urutan Prioritas Penyelesaian Kasus 2.....	31
Gambar 4.10	Kejadian <i>Bottleneck</i> Lainnya pada Kasus 2	31
Gambar 4.11	Model Petri Net dari Jalur Kereta Api Waru-Sidoarjo.....	32
Gambar 4.12	Petri Net dari Permasalahan Kasus 1.....	33
Gambar 4.13	Ilustrasi Kejadian <i>Deadlock</i> dari Kasus 1	34
Gambar 4.14	Ilustrasi Kejadian <i>Deadlock</i> Lainnya dari Kasus 1	35
Gambar 4.15	Model Petri Net dengan Prioritas	36

Gambar 4.16 Ilustrasi Posisi Saat Terjadinya Pertemuan Kereta Api yang Berjalan Berlawanan Arah.....	38
Gambar 4.17 Waktu Kedatangan dan Keberangkatan Kereta Api 400 Menit Pertama	50

DAFTAR TABEL

Tabel 4.1 Jadwal Kedatangan dan Keberangkatan Kereta Api dari Arah WR Menuju ke SDA	51
Tabel 4.2 Jadwal Kedatangan dan Keberangkatan Kereta Api dari Arah SDA Menuju ke WR	53

DAFTAR SIMBOL

$\mathbb{N}$	: Gabungan Himpunan Bilangan Asli dan Nol.
$\mathbb{R}$	: Himpunan bilangan real.
$\mathbb{R}_\epsilon$	: Gabungan Himpunan bilangan real dan $\epsilon = -\infty$ atau ditulis sebagai $\mathbb{R}_\epsilon = \mathbb{R} \cup \{\epsilon\}$.
$\oplus$	: Operasi max atau maksimum (“baca: <i>oplus</i> ”).
$\otimes$	: Operasi + atau penjumlahan (“baca: <i>otimes</i> ”).
$\mathbb{R}_{\max}$	: Struktur aljabar yang terdiri dari himpunan $\mathbb{R}_\epsilon$ dengan dua operasi biner $\oplus$ dan $\otimes$.
ϵ	: Elemen netral dalam Aljabar Max-Plus dengan $\epsilon = -\infty$.
e	: Elemen satuan dalam Aljabar Max-Plus dengan $e = 0$.
$\mathbb{R}_\epsilon^{m \times n}$	: Himpunan matriks yang elemen-elemennya di $\mathbb{R}_\epsilon$.
$a_{i,j}$	: Elemen baris- i kolom- j dari suatu matriks $A \in \mathbb{R}_\epsilon^{m \times n}$.
$[A]_{i,j}$	: Elemen baris- i kolom- j dari matriks $A \in \mathbb{R}_\epsilon^{m \times n}$ dapat dinotasikan pula dengan $a_{i,j}$.
$\underline{n}$	: Himpunan bilangan asli dari 1 sampai dengan n atau ditulis sebagai $\underline{n} = \{1, 2, \dots, n\}$.
$\mathcal{E}(m, n)$	: Matriks berukuran $m \times n$ dengan elemen $[\mathcal{E}]_{i,j} = \epsilon$ dengan $i \in \underline{m}$ dan $j \in \underline{n}$.
$E(m, n)$	: Matriks berukuran $m \times n$ dengan elemen diagonal utama sama dengan e dan elemen yang lainnya sama dengan ϵ .
$\mathbb{R}_\epsilon^m$	: Himpunan vektor yang elemen-elemennya di $\mathbb{R}_\epsilon$.
x_j	: Elemen ke- j dari suatu vektor $\mathbf{x} \in \mathbb{R}_\epsilon^n$.
$[\mathbf{x}]_j$	: Elemen ke- j dari suatu vektor $\mathbf{x} \in \mathbb{R}_\epsilon^n$ atau dapat dinotasikan pula dengan x_j .
$\mathbf{u}$	: Vektor satuan dengan $[\mathbf{u}]_j = e$ untuk $j \in \underline{n}$.
A'	: <i>Transpose</i> dari suatu elemen $A \in \mathbb{R}_\epsilon^{m \times n}$ dengan $[A']_{ij} = a_{ji}$ untuk $i \in \underline{m}$ dan $j \in \underline{n}$.
λ	: Nilai karakteristik (nilai eigen) dari suatu matrik $A \in \mathbb{R}_\epsilon^{n \times n}$.

$\boldsymbol{v}$: Vektor karakteristik (vektor eigen) yang berkaitan dengan nilai eigen λ dari suatu matrik $A \in \mathbb{R}_\epsilon^{n \times n}$.

BAB 1

PENDAHULUAN

Pada bab ini diberikan ulasan mengenai hal-hal yang melatarbelakangi penelitian. Di dalamnya mencakup identifikasi permasalahan, maupun beberapa informasi tentang penelitian terdahulu yang berhubungan dengan topik tesis. Uraian ini bersifat umum, yaitu menjelaskan secara ringkas hal-hal yang dilakukan pada proses penelitian. Dari ulasan tersebut kemudian dirumuskan permasalahan yang dibahas, batasan masalah, tujuan, dan manfaat hasil penelitian.

1.1 Latar Belakang

Transportasi atau pengangkutan merupakan bidang kegiatan yang sangat penting dalam kehidupan masyarakat Indonesia. Pentingnya transportasi bagi masyarakat Indonesia disebabkan oleh beberapa faktor antara lain, keadaan geografis Indonesia yang terdiri dari ribuan pulau kecil dan besar, perairan yang terdiri dari sebagian besar laut, sungai dan danau yang memungkinkan pengangkutan dilakukan melalui darat, perairan, dan udara guna menjangkau seluruh wilayah Indonesia (Abdulkadir, 1998, hal. 7).

Saat ini, pertumbuhan ekonomi Indonesia yang dicapai sampai dengan triwulan IV-2013 masih belum merata. Pertumbuhan lebih banyak terkonsentrasi pada sektor, regional, dan kelompok tertentu. Dari sisi sektor konsentrasinya terletak pada konsumsi, sementara regional, pertumbuhan terkonsentrasi di Pulau Jawa. Dari sisi golongan pendapatan, pertumbuhan hanya dinikmati golongan menengah ke atas. Berdasarkan data Badan Pusat Statistik, struktur perekonomian Indonesia secara spasial masih didominasi Pulau Jawa dengan kontribusi terhadap Produk Domestik Bruto (PDB) sebesar 57,78 persen. Komponen pengeluaran rumah tangga menyumbang 55,82 persen (Badan Pusat Statistik, 2014).

Masih tingginya perbedaan pembangunan antara kota-kota besar dan daerah di sekitarnya juga merupakan penyebab terjadinya transportasi, yaitu berupa pengangkutan barang maupun manusianya itu sendiri (arus urbanisasi). Hingga saat ini, sekitar 80% pergerakan transportasi di Pulau Jawa masih

didominasi oleh transportasi jalan. Para pelaku usaha lebih memilih penggunaan truk daripada kereta api karena alasan *handling*, jadwal, aksesibilitas, dan sebagainya. Untuk Pulau Jawa, jalan merupakan salah satu infrastruktur yang kurang memadai karena tingginya beban. Sehingga berakibat terhadap waktu tempuh yang lama, sehingga biaya transportasi dan logistik menjadi mahal.

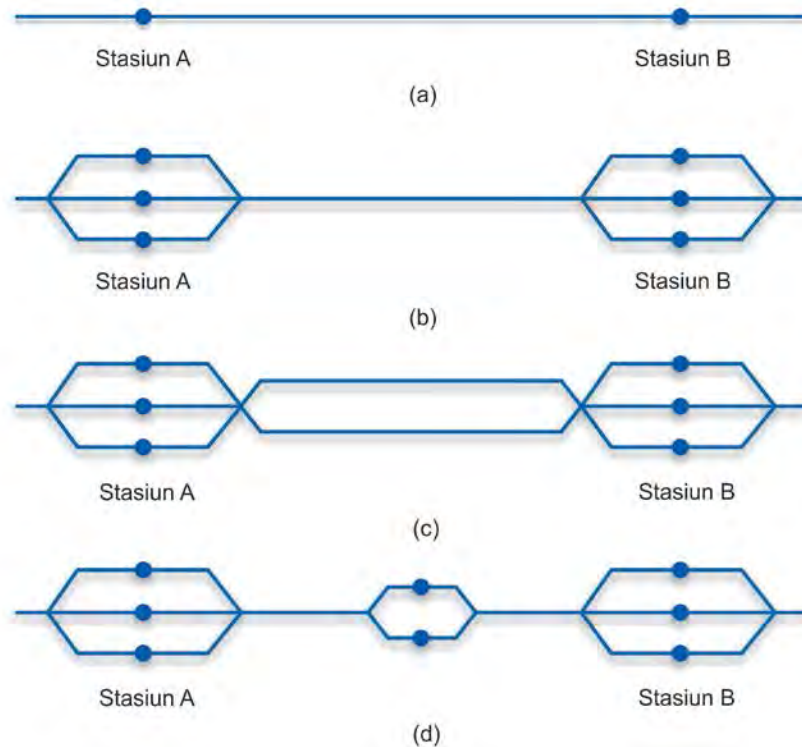
Sedangkan akibat dari adanya arus urbanisasi dari daerah ke kota-kota besar terutama kota Jakarta menyebabkan penduduk makin terpusat di kota sehingga tiap menjelang Lebaran terjadi arus mudik besar-besaran dari kota ke daerah. Permasalahan arus mudik ini mengharuskan pemerintah untuk memperhatikannya secara khusus setiap tahunnya. Itu bisa dilihat dari perbaikan infrastruktur dan pengamatan khusus dengan menempatkan sejumlah CCTV untuk mempermudah dalam pemantauan.

Berdasarkan survey persepsi terhadap pemudik menggunakan angkutan umum di terminal diperoleh kesimpulan bahwa secara umum moda yang digunakan pemudik tahun 2013 adalah Bus (56,5%), Tidak Mudik (20,5%), Pemudik yang menggunakan Kereta Api (6,8%), sedangkan moda yang digunakan Pebalik pengguna angkutan umum adalah Bus (69%), Tidak Mudik (17%), Pebalik yang menggunakan Kereta Api (7%) (Direktorat Jenderal Perhubungan Darat, 2014).

Dalam mengatasi permasalahan transportasi ini pemerintah telah menetapkan beberapa kebijakan, misalnya dengan memperbaiki infrastruktur bangunan jalan, membangun jembatan antar pulau, dan pembangunan jalur kereta api *double track* pada transportasi darat. Pembangunan jalur kereta api *double track* ini masih diterapkan pada Lintas Utara Pulau Jawa (Jakarta-Semarang-Surabaya) dan akan diteruskan pada jalur lainnya (Lintas Selatan Pulau Jawa yaitu Jakarta-Yogyakarta-Surabaya). Saat ini untuk Lintas Utara Pulau Jawa telah selesai pengerjaannya dan untuk Lintas Selatan Pulau Jawa masih terhambat dengan masalah pembebasan lahan.

Ide penelitian ini adalah mengoptimalkan penggunaan jalur kereta api *single track* dengan menambahkan persimpangan di tengah-tengah jalur kereta api *single track* di antara dua stasiun utama yang selanjutnya diberi istilah jalur kereta api *semi-double track*, sebagai ilustrasi bisa dilihat pada Gambar 1.1. Jalur kereta

api *semi-double track* ini banyak dijumpai pada jalur kereta api di Indonesia, sebagai contohnya pada sebagian besar jalur kereta api Lintas Selatan Pulau Jawa.



Gambar 1.1 Ilustrasi Permasalahan dari Dua Stasiun Kereta Api (a) Dua Stasiun yang Digambarkan dalam Sebuah Graf, (b) Ilustrasi Jalur Kereta Api *Single Track*, (c) Ilustrasi Jalur Kereta Api *Double Track*, (d) Ilustrasi Jalur Kereta Api *Semi-Double Track*

Permasalahan pada jalur kereta api *semi-double track* adalah bagaimana menentukan struktur hirarkis arah jalur lintasannya. Hal ini disebabkan pembagian *resource* (rel) secara bergantian untuk beberapa kereta api yang melaluinya. Berbeda dengan jalur kereta api *double track*, kita bisa langsung menentukan arah jalur lintasan yang berbeda-beda untuk setiap *resource*-nya. Permasalahan seperti ini pernah dibahas oleh Danjing Li (2008) dalam disertasinya dengan judul *A Hierarchical Control Structure for A Class of Timed Discrete Event Systems*.

Dari beberapa penjelasan di atas, pada penelitian ini dimaksudkan untuk mencari solusi alternatif dalam mengatasi permasalahan transportasi khususnya jalur kereta api, dengan menggunakan Petri Net dan Aljabar Max-Plus sebagai alat pemodelannya.

1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan di atas, permasalahan yang dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menentukan struktur hirarkis jalur kereta api *semi-double track*?
2. Bagaimana mengkontruksi model Petri Net dari jalur kereta api *semi-double track*?
3. Bagaimana menjabarkan model Petri Net jalur kereta api *semi-double track* ke dalam model Aljabar Max-Plus?
4. Bagaimana sifat keperiodikan dari sistem jalur kereta api *semi-double track* berdasarkan model Aljabar Max-Plus?

1.3 Batasan Masalah

Penelitian ini dilakukan pada jalur kereta api Waru-Sidoarjo, sehingga data yang digunakan adalah data jalur dan jadwal keberangkatan kereta api Waru-Sidoarjo yang diambil dari PT Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya. Data jalur digunakan sebagai landasan membentuk model Petri Net, sedangkan jadwal keberangkatan kereta api digunakan sebagai inputan nilai parameter dalam melakukan perhitungan nilai eigen dari model Aljabar Max-Plus. Jadwal keberangkatan kereta api tersebut terdiri dari kereta api reguler dan kereta api barang. Nilai parameter yang digunakan dalam penelitian ini adalah waktu tunggu dan waktu tempuh kereta api. Dalam hal ini, diambil nilai rata-rata dari data jadwal keberangkatan kereta api tersebut (keterlambatan tidak diperhitungkan).

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut:

1. Memperoleh struktur hirarkis jalur kereta api *semi-double track*.
2. Memperoleh model Petri Net dari jalur kereta api *semi-double track*.
3. Memperoleh model Aljabar Max-Plus dari model Petri Net jalur kereta api *semi-double track*.

4. Mengidentifikasi sifat keperiodikan dari sistem jalur kereta api *semi-double track* berdasarkan model Aljabar Max-Plus.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat memberikan solusi alternatif untuk mengurangi permasalahan transportasi khususnya jalur kereta api dengan menerapkan penggunaan sistem jalur kereta api *semi-double track*, atau dapat dijadikan sebagai bahan pertimbangan dalam menetapkan kebijakan pemerintah mengenai pembangunan jalur kereta api *double track* pada Lintas Selatan Pulau Jawa yang masih dalam tahap perencanaan dan menemui beberapa kendala, salah satunya yaitu masalah pembebasan lahan.

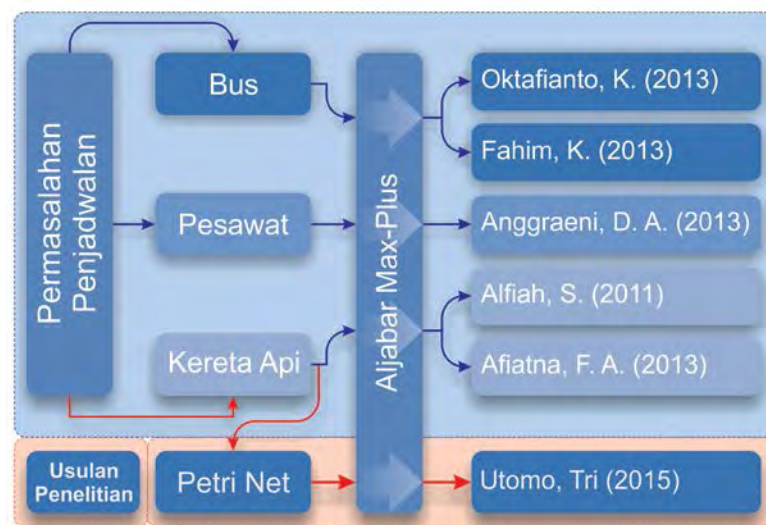
BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini dikaji hal-hal yang berkaitan dengan penelitian-penelitian sebelumnya mengenai permasalahan penjadwalan dan dasar teori yang menjadi dasar dalam menyelesaikan permasalahan penelitian yang akan dilakukan yaitu Petri Net dan Aljabar Max-Plus.

2.1 Kajian Pustaka

Ada beberapa penelitian yang terkait dengan masalah sistem penjadwalan menggunakan Aljabar Max-Plus mulai dari sistem penjadwalan keberangkatan bus, kereta api, dan pesawat, seperti yang dilakukan oleh Kresna Oktafianto (2013) dengan judul *Implementasi Aljabar Max-Plus pada Pemodelan dan Penjadwalan Keberangkatan Bus Kota Damri (Studi Kasus di Surabaya)*. Dalam penelitian ini menghasilkan kesimpulan bahwa periode keberangkatan bus kota Damri di masing-masing halte adalah 6 menit. Selain itu Kistosil Fahim (2013) dengan judul *Aplikasi Aljabar Max-Plus pada Pemodelan dan Penjadwalan Busway yang Diintegrasikan dengan Kereta Api Komuter*. Penelitian ini menghasilkan kesimpulan bahwa periode keberangkatan setiap λ menit sekali, dengan $41,6 < \lambda < 51,9$.



Gambar 2.1 Penelitian Terdahulu

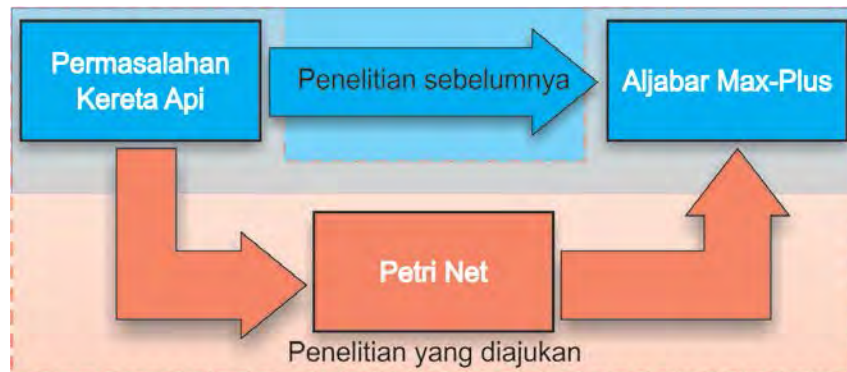
Penelitian terkait dengan penjadwalan pesawat dilakukan oleh Dyah Arum Anggraeni (2013) dengan judul *Analisis Jadwal Keberangkatan Pesawat Transit di Bandara dengan Menggunakan Aljabar Max-Plus*. Sedangkan penelitian terkait dengan penjadwalan kereta api juga sudah pernah dilakukan oleh Fatma Ayu Nuning Farida Afiatna (2013) yaitu *Pemodelan dan Penjadwalan Monorel dan Trem yang Terintegrasi di Kota Surabaya Menggunakan Aljabar Max-Plus*, dan menghasilkan kesimpulan periode keberangkatannya adalah 4,6. Selain itu Siti Alfiah (2011) dengan judul *Pemodelan dan Penjadwalan Jaringan Kereta Rel Listrik (KRL) Menggunakan Aljabar Max-Plus* juga meneliti tentang penjadwalan kereta api. Akan tetapi penelitian-penelitian mengenai penjadwalan kereta api ini hanya membahas penjadwalannya tanpa melihat jalur kereta yang digunakan yaitu *single track* atau *double track*. Sehingga peneliti ingin melakukan penelitian terkait masalah penjadwalan kereta api dengan mempertimbangkan jalur kereta api yang digunakan dengan memanfaatkan Petri Net sebagai alat pemodelannya (lihat Gambar 2.1 dan Gambar 2.2).

Ide penelitian ini adalah mengoptimalkan penggunaan jalur kereta api *single track* dengan menambahkan persimpangan di tengah-tengah jalur kereta api *single track* di antara dua stasiun utama yang selanjutnya diberi istilah jalur kereta api *semi-double track*, sebagai ilustrasi bisa dilihat pada Gambar 1.1. Jalur kereta api *semi-double track* ini banyak dijumpai pada jalur kereta api di Indonesia, sebagai contohnya pada sebagian besar jalur kereta api Lintas Selatan Pulau Jawa.

Permasalahan pada jalur kereta api *semi-double track* adalah bagaimana menentukan struktur hirarkis arah jalur lintasannya. Hal ini disebabkan pembagian *resource* (rel) secara bergantian untuk beberapa kereta api yang melaluinya. Berbeda dengan jalur kereta api *double track*, kita bisa langsung menentukan arah jalur lintasan yang berbeda-beda untuk setiap *resource*-nya. Permasalahan serupa pernah dibahas oleh Danjing Li (2008) dalam disertasinya dengan judul *A Hierarchical Control Structure for A Class of Timed Discrete Event Systems*, dalam hal ini yang menjadi sumber rujukan utama penelitian.

Penelitian ini dilakukan pada jalur kereta api Waru-Sidoarjo, kemudian dengan menggambarkan permasalahan kereta api seperti pada Gambar 1.1, nantinya akan dilakukan pembentukan model menggunakan Petri Net dan

memberikan struktur hirarkis arah jalur lintasannya, kemudian dijabarkan ke dalam model Aljabar Max-Plus.



Gambar 2.2 Ilustrasi Alur Penyelesaian Permasalahan

2.2 Petri Net

Dalam proses penelitian ini, terdapat beberapa teori yang berkaitan dengan Petri Net. Beberapa teori tersebut meliputi definisi dan notasi dalam Petri Net, Petri Net bertanda dan ruang keadaan, dan dinamika Petri Net.

2.2.1 Definisi dan Notasi dalam Petri Net

Petri Net adalah suatu graf bipartisi, yang terdiri dari dua himpunan bagian P dan T , masing-masing menyatakan *place* dan *transition*. Petri Net merupakan salah satu alat yang biasa digunakan untuk pemodelan dan penjadwalan matematika dan mudah diaplikasikan untuk banyak sistem, misalnya untuk menjelaskan dan mendeskripsikan informasi (proses dari suatu sistem).

Secara matematis Petri Net dapat dituliskan sebagai 4-tuple (P, T, A, w) dengan:

$P = \{p_1, p_2, \dots, p_m\}$ adalah himpunan berhingga dari *places*,

$T = \{t_1, t_2, \dots, t_n\}$ adalah himpunan berhingga dari *transition*,

$A \subseteq (P \times T) \cup (T \times P)$ adalah himpunan dari garis berarah (*arcs*),

$w: A \rightarrow \{1, 2, 3, \dots\}$ adalah fungsi bobot pada *arcs*,

diasumsikan bahwa pada (P, T, A, w) tidak ada *place* dan *transition* yang terisolasi (Cassandras & Lafortune, 2008).

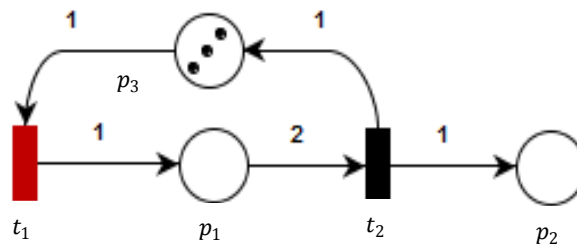
Dalam menjelaskan Petri Net, digunakan notasi $I(t_j)$ untuk merepresentasikan himpunan *place* input dari *transition* t_j , dan notasi $O(t_j)$ untuk merepresentasikan himpunan *place* output dari *transition* t_j . Secara matematis ditulis,

$$I(t_j) = \{p_i \in P: (p_i, t_j) \in A\}$$

$$O(t_j) = \{p_i \in P: (t_j, p_i) \in A\}$$

Notasi yang sama dapat digunakan untuk merepresentasikan *transition* input dan output dari *place* p_i , yaitu $I(p_i)$ dan $O(p_i)$ (Cassandras & Lafortune, 2008).

Dalam menggambar Petri Net, tipe dari sebuah titik akan dibedakan menjadi dua (*place* dan *transition*). Dalam tesis ini digunakan lingkaran untuk merepresentasikan sebuah *place* dan persegi panjang untuk merepresentasikan sebuah *transition*. *Arcs* yang menghubungkan *place* dan *transition* merepresentasikan elemen dari himpunan A . *Arc* yang menghubungkan *place* p_i ke *transition* t_j berarti $p_i \in I(t_j)$. Jika bobot *arc* dari *place* p_i ke *transition* t_j adalah k ditulis $w(p_i, t_j) = k$ maka terdapat *arc* sebanyak k dari *place* p_i ke *transition* t_j atau sebuah *arc* dengan bobot k .



Gambar 2.3 Contoh Petri Net Sederhana

Dari contoh Petri Net sederhana pada Gambar 2.3 diketahui bahwa terdapat tiga *place* yaitu $P = \{p_1, p_2, p_3\}$, dua *transition* yaitu $T = \{t_1, t_2\}$, dan 5 *arcs* yaitu $A = \{(t_1, p_1), (p_1, t_2), (t_2, p_2), (t_2, p_3), (p_3, t_1)\}$ dengan masing-masing bobot dari setiap *arc* tersebut adalah $w(t_1, p_1) = 1$, $w(p_1, t_2) = 2$, $w(t_2, p_2) = 1$, $w(t_2, p_3) = 1$, dan $w(p_3, t_1) = 1$.

Sedangkan untuk input dan output dari masing-masing *place* dan *transition* adalah $I(p_1) = \{t_1\}$, $O(p_1) = \{t_2\}$, $I(p_2) = \{t_2\}$, $I(p_3) = \{t_2\}$, $O(p_3) = \{t_1\}$, $O(t_1) = \{p_1\}$, $O(t_1) = \{p_1\}$, $I(t_2) = \{p_1\}$, dan $O(t_2) = \{p_2, p_3\}$.

2.2.2 Petri Net Bertanda dan Ruang Keadaan

Transitions pada Petri Net merupakan representasi dari pergerakan *event* pada Sistem Event Diskrit (SED), sedangkan *place* merepresentasikan kondisi agar *event* pada sistem bisa terjadi. Dalam hal ini, diperlukan alat sebagai penanda apakah kondisi tersebut dapat terpenuhi atau tidak, dengan cara memberikan *token* yaitu suatu tanda yang diletakkan pada *place*. Dalam model Petri Net, *token* direpresentasikan dengan *dot*, dan jika jumlah *token* banyak maka dituliskan dengan angka.

Penanda (*marking*) x pada Petri Net adalah fungsi $x: P \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$. Penanda dinyatakan dengan vektor kolom yang elemen-elemennya berkaitan dengan x , yaitu bilangan bulat tak negatif (banyaknya *token* dalam suatu *place*), atau dapat ditulis sebagai $x = [x(p_1), x(p_2), \dots, x(p_m)]'$. Elemen ke- i pada vektor ini merupakan banyaknya *token* pada *place* p_i , dengan $x(p_i) \in \{0, 1, 2, \dots\}$. Dalam notasi matematika Petri Net bertanda ditulis sebagai 5-tuple (P, T, A, w, x_0) dengan (P, T, A, w) adalah Petri Net dan x_0 adalah Penanda awal. Pada Gambar 2.3 menunjukkan bahwa $x_0 = [0 \ 0 \ 3]'$. Untuk selanjutnya Petri Net bertanda cukup disebut sebagai Petri Net saja (Subiono, 2015).

Suatu *transition* $t_j \in T$ dalam sebuah Petri Net dikatakan kondisinya terpenuhi (*enabled*) jika

$$x(p_i) \geq w(p_i, t_j) \text{ untuk setiap } p_i \in I(t_j)$$

(Cassandras & Lafortune, 2008).

Dengan kata lain, *transition* t_j dalam sebuah Petri Net *enabled* jika jumlah *token* pada *place* p_i sekurang-kurangnya sama dengan bobot pada *arc* yang menghubungkan p_i dengan t_j , untuk semua p_i yang menjadi input dari *transition* t_j .

Pada Gambar 2.3 *transition* t_1 merupakan *transition* yang *enabled*, karena *token* pada *place* p_3 yang menjadi input dari *transition* t_1 lebih besar dari bobot *arc* yang menghubungkan *place* p_3 dengan *transition* t_1 . Sedangkan *transition* t_2 tidak *enabled* karena *token* pada *place* p_1 yang menjadi input dari *transition* t_2 lebih kecil dari bobot *arc* yang menghubungkan *place* p_1 dengan *transition* t_2 .

2.2.3 Dinamika Petri Net

Mekanisme perubahan keadaan pada Petri Net ditandai dengan perpindahan *token-token* pada Petri Net tersebut sehingga mengubah keadaan Petri Net. Secara umum, sebuah *transition* pada Petri Net yang *enabled*, diistilahkan dengan dapat di-*fire* (*firing* diartikan dengan *transition* di-*fire*). Fungsi perubahan keadaan pada Petri Net ini merupakan perubahan keadaan Petri Net sebelum dan sesudah suatu *transition* di-*fire*.

Suatu fungsi perubahan keadaan pada sebuah Petri Net (P, T, A, w, x) , yaitu $f: \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$ terdefinisi untuk *transition* $t_j \in T$ jika dan hanya jika

$$x(p_i) \geq w(p_i, t_j) \text{ untuk semua } p_i \in I(t_j).$$

Jika $f(x, t_j)$ terdefinisi, maka ditulis $y = f(x, t_j)$ dengan elemen-elemen dari y adalah $y(p_i)$ didapatkan dari

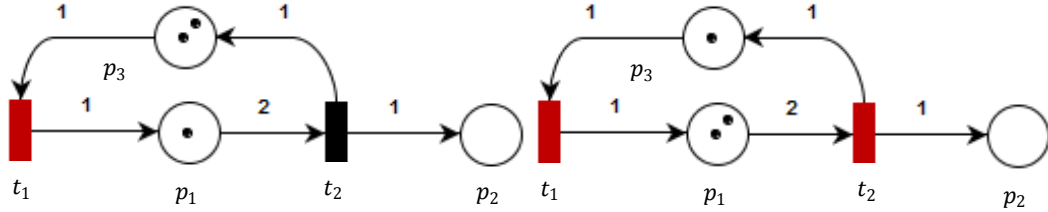
$$y(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i), i = 1, 2, \dots, m$$

(Cassandras & Lafortune, 2008).

Dengan kata lain, hanya *transition enabled* yang dapat di-*fire*. Proses yang terjadi pada *firing transition* menyebabkan semua *token* pada *place* input berkurang sebanyak bobot *arc* yang menghubungkannya dengan *transition* yang di-*fire* dan *token* pada *place* output bertambah sebanyak bobot *arc* yang menghubungkannya dengan *transition* yang di-*fire*.

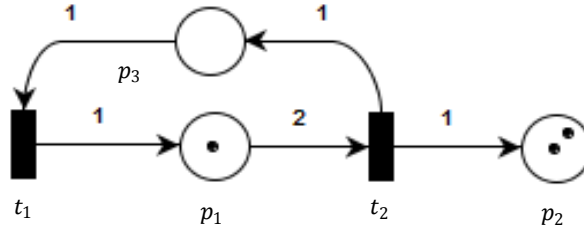
Pada contoh Gambar 2.3, terlihat bahwa *transition* yang *enabled* adalah t_1 saja dan keadaan awal Petri Net tersebut adalah $x_0 = [0 \ 0 \ 3]'$. Jika t_1 di-*fire*, maka keadaan Petri Net berikutnya adalah $x_1 = [1 \ 0 \ 2]'$, dan *transition* yang *enabled* tetap hanya t_1 saja (lihat Gambar 2.4 sebelah kiri). Jika selanjutnya

transition t_1 di-*fire* untuk yang kedua kalinya, maka keadaan Petri Net berikutnya akan menjadi $x_2 = [2 \ 0 \ 1]'$, dan *transition* yang *enabled* adalah t_1 dan t_2 (Gambar 2.4 sebelah kanan), dan seterusnya.



Gambar 2.4 Keadaan Setelah t_1 di-*fire* yang Pertama (Kiri) dan Kedua (Kanan)

Sebagai tambahan informasi, jika semua *transition* pada sebuah Petri Net tidak ada yang *enabled* maka keadaan tersebut dinamakan dengan *deadlock*, dan model Petri Net yang baik adalah yang bisa menghindari adanya kejadian *deadlock*. Contoh kejadian *deadlock* bisa dilihat pada Gambar 2.3, jika Petri Net tersebut di-*fire* 7 kali berturut-turut pada *transition* $t_1, t_1, t_1, t_2, t_1, t_2, t_1$, maka petrinet tersebut dalam keadaan *deadlock*, dengan akhir keadaan Petri Net $x_7 = [1 \ 2 \ 0]'$ (Lihat Gambar 2.5).



Gambar 2.5 Contoh Keadaan *Deadlock*

2.3 Aljabar Max-Plus

Dalam proses penelitian ini juga terdapat beberapa teori yang berkaitan dengan Aljabar Max-Plus. Beberapa teori tersebut meliputi definisi dan notasi dalam Aljabar Max-Plus, vektor dan matriks atas Aljabar Max-Plus, nilai eigen dan vektor eigen, dan solusi persamaan linear atas Aljabar Max-Plus.

2.3.1 Definisi dan Notasi dalam Aljabar Max-Plus

Aljabar Max-Plus adalah suatu struktur aljabar yang terdiri dari $\mathbb{R}_\epsilon$ dengan $\mathbb{R}_\epsilon = \mathbb{R} \cup \{\epsilon\}$ dan $\epsilon = -\infty$ dengan dua operator biner yaitu Operator Max

$(\oplus$ “baca: *oplus*”) dan Operator Plus ($\otimes$ “baca: *otimes*”), yang didefinisikan sebagai berikut, $\forall x, y \in \mathbb{R}_\epsilon$

$$x \oplus y = \max\{x, y\} \text{ dan } x \otimes y = x + y.$$

Diketahui bahwa $(\mathbb{R}_\epsilon, \oplus, \otimes)$ merupakan semiring komutatif dengan elemen netral ϵ dan elemen satuan e ($e = 0$). Lebih lanjut lagi, $(\mathbb{R}_\epsilon, \oplus, \otimes)$ merupakan semifield idempoten (Subiono, 2015).

Seperti halnya dalam Aljabar Biasa (*conventional algebra*), prioritas urutan operasi $\otimes$ lebih dulu atas operasi $\oplus$. Begitu juga dengan elemen-elemen pada $\mathbb{R}_\epsilon$ disebut juga sebagai skalar. Untuk keperluan dalam penulisan selanjutnya, penulisan struktur Aljabar Max-Plus $(\mathbb{R}_\epsilon, \oplus, \otimes)$ cukup ditulis sebagai $\mathbb{R}_{\max}$.

Operasi pangkat dalam Aljabar Max-Plus diperkenalkan dengan menggunakan sifat asosiatif. Gabungan dari Himpunan Bilangan Asli dan Nol dinotasikan dengan $\mathbb{N}$, didefinisikan untuk $x \in \mathbb{R}_\epsilon$ maka,

$$x^{\otimes n} = \underbrace{x \otimes x \otimes \dots \otimes x}_{n \text{ kali}}$$

untuk setiap $n \in \mathbb{N}$ dengan $n \neq 0$, dan untuk $n = 0$ didefinisikan sebagai $x^{\otimes n} = e$. Amati bahwa $x^{\otimes n}$, untuk setiap $n \in \mathbb{N}$, kita dapat menerjemahkannya ke dalam Aljabar Biasa ditulis sebagai berikut,

$$x^{\otimes n} = \underbrace{x \otimes x \otimes \dots \otimes x}_{n \text{ kali}} = \underbrace{x + x + \dots + x}_{n \text{ kali}} = n \times x.$$

Terinspirasi dari pengertian ini, dengan cara serupa dapat dilakukan operasi pangkat negatif dari bilangan real dan lebih lanjut mengenai akar-akar dalam Aljabar Max-Plus didefinisikan sebagai berikut, untuk $x \in \mathbb{R}_\epsilon$ dan $x \neq \{\epsilon\}$ maka,

$$x^{\otimes \alpha} = \alpha \times x$$

untuk setiap $\alpha \in \mathbb{R}$.

2.3.2 Vektor dan Matriks atas Aljabar Max-Plus

Dijelaskan dalam bukunya (Heidergott dkk, 2006) bahwa, himpunan matriks berukuran $m \times n$ dalam Aljabar Max-Plus dinotasikan dengan $\mathbb{R}_\epsilon^{m \times n}$. Untuk $n \in \mathbb{N}$ dan $n \neq 0$, didefinisikan $\underline{n} = \{1, 2, \dots, n\}$. Elemen matriks $A \in \mathbb{R}_\epsilon^{m \times n}$ pada baris ke- i dan pada kolom ke- j dinotasikan dengan a_{ij} , untuk $i \in \underline{m}$ dan $j \in \underline{n}$. Dalam hal ini matriks A mempunyai m baris dan n kolom dan dapat ditulis sebagai berikut,

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

Ada kalanya elemen a_{ij} dinotasikan sebagai $[A]_{ij}$, dengan $i \in \underline{m}$ dan $j \in \underline{n}$.

Operasi penjumlahan matriks $A, B \in \mathbb{R}_\epsilon^{m \times n}$ atau ditulis $A \oplus B$ didefinisikan sebagai berikut,

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max\{a_{ij}, b_{ij}\}.$$

Diketahui bahwa $A \oplus B = B \oplus A$, karena $[A \oplus B]_{ij} = \max\{a_{ij}, b_{ij}\} = \max\{b_{ij}, a_{ij}\} = [B \oplus A]_{ij}$.

Operasi perkalian matriks $A \in \mathbb{R}_\epsilon^{m \times n}$ dengan skalar $\alpha \in \mathbb{R}_\epsilon$ atau ditulis $\alpha \otimes A$ didefinisikan sebagai berikut,

$$[\alpha \otimes A]_{ij} = \alpha \otimes a_{ij} \text{ dengan } i \in \underline{m} \text{ dan } j \in \underline{n}.$$

Sedangkan operasi perkalian matriks $A \in \mathbb{R}_\epsilon^{m \times l}$ dan $B \in \mathbb{R}_\epsilon^{l \times n}$ atau ditulis $A \otimes B$ didefinisikan sebagai berikut,

$$[A \otimes B]_{ik} = \bigoplus_{j=1}^l a_{ij} \otimes b_{jk} = \max_{j \in \underline{l}} \{a_{ij} + b_{jk}\}$$

untuk setiap $i \in \underline{m}$, dan $k \in \underline{n}$. Operasi perkalian matriks atas $\mathbb{R}_{\max}$ ini serupa dengan operasi perkalian pada Aljabar Biasa, yaitu dengan mengganti operasi $+$ dengan $\oplus$ dan mengganti operasi $\times$ dengan operasi $\otimes$.

Seperti halnya dalam Aljabar Biasa, operasi perkalian matriks atas $\mathbb{R}_{\max}$ ini juga tidak bersifat komutatif, dengan kata lain

$$A \otimes B \neq B \otimes A.$$

Vektor adalah bentuk khusus dari matrik, yaitu $\mathbb{R}_\epsilon^{m \times n}$ dengan $n = 1$, atau biasa dinotasikan dengan $\mathbb{R}_\epsilon^m$. Element ke- j dari sebuah vektor $\mathbf{x} \in \mathbb{R}_\epsilon^m$, dinotasikan dengan x_j atau $[\mathbf{x}]_j$. Vektor yang elemen-elemennya adalah e dinamakan vektor satuan (*unit vector*) dinotasikan dengan $\mathbf{u}$. Secara matematis ditulis $[\mathbf{u}]_j = e$ untuk $j \in \underline{m}$ (Heidergott dkk, 2006).

Sebagai tambahan informasi, didefinisikan matriks $\mathcal{E}(m, n)$ menyatakan matriks berukuran $m \times n$ dengan semua elemen pada $\mathcal{E}$ sama dengan ϵ dinamakan matrik nol (*null matrix*), dan matriks $E(m, n)$ menyatakan matriks berukuran $m \times n$ yang didefinisikan sebagai

$$[E(m, n)]_{ij} = f(x) = \begin{cases} e, & i = j \\ \epsilon, & i \neq j \end{cases}$$

jika $m = n$, maka matriks $E(m, m)$ dinamakan matriks identitas (*identity matrix*). Untuk selanjutnya, jika dimensi ukuran dari $\mathcal{E}(m, n)$ dan $E(m, n)$ sudah jelas, maka cukup ditulis dengan $\mathcal{E}$ dan E . Dalam bukunya (Subiono, 2015) menjelaskan bahwa untuk $(\mathbb{R}_\epsilon^{m \times m}, \oplus, \otimes)$ merupakan semiring idempoten dengan elemen nol $\mathcal{E}$ dan elemen satuan E .

Transpose dari suatu elemen $A \in \mathbb{R}_\epsilon^{m \times n}$ dinotasikan dengan A' didefinisikan sebagai $[A']_{ij} = a_{ji}$ untuk $i \in \underline{m}$ dan $j \in \underline{n}$. Selain itu untuk $A \in \mathbb{R}_\epsilon^{m \times m}$, diperkenalkan pangkat ke- k dari A yang dinotasikan $A^{\otimes k}$ didefinisikan sebagai

$$A^{\otimes k} = \underbrace{A \otimes A \otimes \dots \otimes A}_{k \text{ kali}}$$

dengan $k \in \mathbb{N}$, dan $k \neq 0$, untuk $k = 0$ didefinisikan $A^{\otimes k} = E(m, m)$.

2.3.3 Nilai Eigen dan Vektor Eigen

Nilai eigen dan vektor eigen merupakan nilai karakteristik dan vektor karakteristik yang bersesuaian dengan matriks persegi A berukuran $m \times m$, yang didefinisikan sebagai berikut, misalkan $A \in \mathbb{R}_\epsilon^{m \times m}$ merupakan matriks persegi. Jika $\lambda \in \mathbb{R}$ adalah skalar dan $\mathbf{v} \in \mathbb{R}_\epsilon^m$ adalah vektor yang setidaknya memuat satu elemen tak nol (ϵ) sedemikian hingga

$$A \otimes \mathbf{v} = \lambda \otimes \mathbf{v}$$

maka λ disebut dengan nilai eigen dari A dan $\mathbf{v}$ disebut vektor eigen dari A yang berkaitan dengan nilai eigen λ (Heidergott dkk, 2006).

Salah satu cara untuk menghitung nilai eigen dan vektor eigen dapat digunakan Algoritma Power (Subiono, 2015) sebagai berikut,

1. Ambil sebarang vektor awa $\mathbf{x}(0) \in \mathbb{R}_\epsilon^m$ dengan $\mathbf{x}(0) \neq \mathbf{0}$.
2. Iterasi persamaan $\mathbf{x}(k+1) = A \otimes \mathbf{x}(k)$ sehingga didapatkan bilangan bulat $p > q \geq 0$ dan bilangan real c sedemikian hingga

$$\mathbf{x}(p) = c \otimes \mathbf{x}(q).$$

3. Hitung nilai eigen dengan menggunakan persamaan berikut

$$\lambda = \frac{c}{p - q}.$$

4. Hitung vektor eigen dengan menggunakan persamaan berikut

$$\mathbf{v} = \bigoplus_{i=1}^{p-q} \left(\lambda^{\otimes (p-q-i)} \otimes \mathbf{x}(q+i-1) \right).$$

2.3.4 Solusi Persamaan Linear

Untuk suatu matriks persegi $A \in \mathbb{R}_\epsilon^{n \times n}$, dapat dibuat matriks A^+ yang didefinisikan sebagai

$$A^+ = \bigoplus_{k=1}^{\infty} A^{\otimes k} \tag{2.1}$$

Elemen $[A^{\otimes k}]_{i,j}$ merupakan bobot maksimum dari semua *path* dengan panjang k dari titik j ke titik i , sehingga $[A^+]_{i,j}$ merupakan bobot maksimum dari

semua *path* dari titik j ke titik i , dengan kata lain dapat disimpulkan sebagai berikut,

$$[A^+]_{i,j} = \max \left\{ [A^{\otimes k}]_{i,j} : k \geq 1 \right\}.$$

Teorema 2.1. (Subiono, 2015) Misalkan $A \in R_\epsilon^{n \times n}$ sedemikian hingga setiap sirkuit di $\mathcal{G}(A)$ mempunyai bobot sirkuit rata-rata kurang dari atau sama dengan e . $\mathcal{G}(A)$ adalah graf komunikasi dari matriks A . Maka berlaku

$$A^+ = \bigoplus_{k=1}^{\infty} A^{\otimes k} = A \otimes A^{\otimes 2} \otimes A^{\otimes 3} \otimes \dots \otimes A^{\otimes n} \in R_\epsilon^{n \times n} \quad (2.2)$$

Bukti. Karena A berukuran $n \times n$, maka semua *path* di $\mathcal{G}(A)$ dari j ke i dengan panjang lebih besar dari n merupakan tersusun dari setidaknya satu sirkuit dan satu *path* dari j ke i dengan panjang setidaknya n . Disini lain, sirkuit pada $\mathcal{G}(A)$ mempunyai bobot tak positif, sehingga

$$[A^+]_{i,j} \leq \max \left\{ [A^{\otimes k}]_{i,j} : k \in \underline{n} \right\}$$

Hal ini menunjukkan bahwa

$$A^+ = A \otimes A^{\otimes 2} \otimes A^{\otimes 3} \otimes \dots \otimes A^{\otimes n} \quad \blacksquare$$

Berdasarkan Persamaan (2.1) dan (2.2), untuk sebarang $A \in R_\epsilon^{n \times n}$ didefinisikan suatu matriks A^* sebagai berikut

$$A^* = E \oplus A^+ = \bigoplus_{k \geq 0} A^{\otimes k} \quad (2.3)$$

Dari Persamaan (2.2) jelas bahwa A^* ada untuk setiap A dengan graf $\mathcal{G}(A)$ hanya mempunyai bobot sirkuit tak-positif. Catatan bahwa $A^{\otimes n}$ merupakan bobot maksimum dari *path* dengan panjang n . Jadi *path* ini memuat setidaknya satu sirkuit. Jika semua sirkuit mempunyai bobot tak positif, maka

$$[A^{\otimes n}]_{i,j} \leq \bigotimes_{k=0}^{n-1} [A^{\otimes k}]_{i,j}, i, j \in \underline{n} \quad (2.4)$$

Dalam kondisi yang sama pada Persamaan (2.2), A^* dapat ditentukan dengan menggunakan persamaan berikut,

$$A^* = \bigoplus_{k=0}^{n-1} A^{\otimes k} \quad (2.5)$$

Untuk selanjutnya notasi ini digunakan untuk menyelesaikan persamaan $\mathbf{x} = (A \otimes \mathbf{x}) \oplus \mathbf{b}$, dengan $A \in R_\epsilon^{n \times n}$ dan $\mathbf{b} \in R_\epsilon^n$.

Teorema 2.2. (Heidergott dkk, 2006) Misalkan $A \in R_\epsilon^{n \times n}$ dan $\mathbf{b} \in R_\epsilon^n$, dan jika bobot sirkuit rata-rata dari $\mathcal{G}(A)$ adalah kurang dari atau sama dengan e , maka $\mathbf{x} = A^* \otimes \mathbf{b}$ adalah solusi dari persamaan $\mathbf{x} = (A \otimes \mathbf{x}) \oplus \mathbf{b}$. $\mathcal{G}(A)$ adalah graf komunikasi dari matriks A . Lebih lanjut, jika bobot sirkuit rata-rata dari $\mathcal{G}(A)$ adalah negatif, maka solusi ini adalah tunggal.

Bukti. Untuk permasalahan pertama dapat dibuktikan dengan cara menunjukkan bahwa

$$A^* \otimes \mathbf{b} = A \otimes (A^* \otimes \mathbf{b}) \oplus \mathbf{b}$$

Berdasarkan Persamaan (2.2) dan penjelasan dari Persamaan (2.3), dapat disimpulkan bahwa A^* ada, sehingga didapat

$$\begin{aligned} A^* \otimes \mathbf{b} &= \bigoplus_{k=0}^{\infty} A^{\otimes k} \otimes \mathbf{b} \\ &= \left(\bigoplus_{k=1}^{\infty} A^{\otimes k} \otimes \mathbf{b} \right) \oplus (E \otimes \mathbf{b}) \\ &= A \otimes \left(\bigoplus_{k=0}^{\infty} A^{\otimes k} \otimes \mathbf{b} \right) \oplus (E \otimes \mathbf{b}) \\ &= A \otimes (A^* \otimes \mathbf{b}) \oplus \mathbf{b} \end{aligned} \quad (2.6)$$

Sampai sini sudah membuktikan sebagian dari teorema di atas. Selanjutnya untuk membuktikan bahwa solusi dari $\mathbf{x} = (A \otimes \mathbf{x}) \oplus \mathbf{b}$ adalah tunggal, pertama dengan mengandaikan $\mathbf{x}$ adalah solusi dari persamaan $\mathbf{x} = \mathbf{b} \oplus (A \otimes \mathbf{x})$,

kemudian substitusikan $\mathbf{x}$ ke dalam persamaan $\mathbf{x} = \mathbf{b} \oplus (A \otimes \mathbf{x})$ sehingga menjadi

$$\mathbf{x} = \mathbf{b} \oplus (A \otimes \mathbf{b}) \oplus (A^{\otimes 2} \otimes \mathbf{x})$$

dan ulangi proses ini sehingga didapatkan

$$\begin{aligned} \mathbf{x} &= \mathbf{b} \oplus (A \otimes \mathbf{b}) \oplus (A^{\otimes 2} \otimes \mathbf{b}) \oplus (A^{\otimes 3} \otimes \mathbf{x}) \\ &= \mathbf{b} \oplus (A \otimes \mathbf{b}) \oplus \dots \oplus (A^{\otimes (k-1)} \otimes \mathbf{b}) \oplus (A^{\otimes k} \otimes \mathbf{x}) \\ &= \bigoplus_{l=0}^{k-1} (A^{\otimes l} \otimes \mathbf{b}) \oplus (A^{\otimes k} \otimes \mathbf{x}) \end{aligned} \quad (2.7)$$

Elemen-elemen $A^{\otimes k}$ adalah bobot maksimum dari *path* dengan panjang k . Untuk k yang cukup besar, setiap *path* memuat satu atau lebih sirkuit elementer turunan sebagai sub-*path*, dan untuk k menuju ke ∞ , maka banyaknya sirkuit elementer yang terjadi juga menuju ke ∞ . Karena sirkuit pada $\mathcal{G}(A)$ mempunyai bobot negatif, maka elemen-elemen dari $A^{\otimes k}$ menuju ke ϵ , seiring k menuju ke ∞ , yaitu

$$\lim_{k \rightarrow \infty} A^{\otimes k} \otimes \mathbf{x} = \epsilon$$

Jadi untuk k menuju ke ∞ , maka Persamaan (2.7) menjadi $\mathbf{x} = A^* \otimes \mathbf{b}$ dalam hal ini

$$\lim_{k \rightarrow \infty} \bigoplus_{l=0}^{k-1} (A^{\otimes l} \otimes \mathbf{b}) = \lim_{k \rightarrow \infty} \bigoplus_{l=0}^{k-1} (A^{\otimes l}) \otimes \mathbf{b} = A^* \otimes \mathbf{b}$$

■

2.4 Persamaan Relasi Rekurensi

Misalkan, τ_{ji} menunjukkan waktu tunggu pada *place* di antara *transition* t_i dan t_j , dalam hal ini hanya ada satu *place*, dan didefinisikan $a_{ji} = \tau_{ji}$ adalah konstan atau tidak bergantung pada waktu k . Dimisalkan juga $x_j(k)$ menunjukkan waktu pada *transition* t_j di-*fire* saat ke- k , sehingga dapat didefinisikan suatu vektor $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_{|T|}(k)]'$ sebagai ruang keadaan sistem.

Untuk sebarang bentuk model Petri Net, dapat dibentuk matriks $A_0, A_1, \dots, A_M$ yang saling berkaitan dengan ukuran $|T| \times |T|$. Untuk mendapatkan $[A_m]_{ji}$ tunjukkan bahwa semua *place* di antara *transition* t_i dan t_j mempunyai inisial token sebanyak m , dan mengambil nilai maksimum dari $[A_m]_{ji}$ dari waktu tunggu yang berkaitan. Hal ini dimungkinkan ada lebih dari satu *place* di antara sebarang dua *transition*, sehingga $[A_m]_{ji} = a_{ji}$, dengan a_{ji} adalah nilai terbesar dari waktu tunggu pada semua *place* di antara *transition* t_i dan t_j sebanyak m token. Jika hanya ada satu *place* maka,

$$[A_m]_{ji} = \begin{cases} a_{ji}, & \text{jika jumlah token pada place } p_{t_j t_i} \text{ adalah } m \\ \epsilon, & \text{selainnya.} \end{cases}$$

Untuk $m = 0, 1, 2, \dots, M$, dengan M adalah jumlah token maksimum yang berkaitan pada *place*.

Dalam bukunya (Heidergott dkk, 2006) menjelaskan bahwa, Jika $x_i(k)$ dianggap sebagai variable keadaan yang menunjukkan waktu *transition* t_i di-fire saat ke- k , dan vektor $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_{|T|}(k)]'$, maka memenuhi kaedah persamaan linear dalam Aljabar Max-Plus (persamaan *implicit* relasi rekurensi).

$$\mathbf{x}(k) = \bigoplus_{m=0}^M A_m \otimes \mathbf{x}(k - m), k \geq 0 \quad (2.8)$$

Persamaan relasi rekurensi di atas merupakan relasi rekurensi orde- M . Persamaan tersebut dapat ditransformasikan ke dalam relasi rekurensi orde-1 seperti berikut ini.

$$\mathbf{x}(k + 1) = A \otimes \mathbf{x}(k), k \geq 0 \quad (2.9)$$

Perlu ditunjukkan terlebih dahulu bahwa A_0 pada Persamaan (2.8) mempunyai bobot sirkuit rata-rata kurang dari atau sama dengan e dengan kata lain tidak mempunyai sirkuit sama sekali. Berdasarkan Teorema 2.1, jika A_0 mempunyai sirkuit dengan bobot kurang dari atau sama dengan e , maka

$$A_0^* = \bigoplus_{i=0}^{n-1} A_0^{\otimes i} \quad (2.10)$$

Kemudian dengan sedikit manipulasi aljabar, yaitu dengan memisalkan bahwa,

$$\mathbf{b}(k) = \bigoplus_{m=0}^M A_m \otimes \mathbf{x}(k-m)$$

Maka Persamaan (2.8) dapat dikonstruksi ulang menjadi persamaan berikut,

$$\mathbf{x}(k) = A_0 \otimes \mathbf{x}(k) \oplus \mathbf{b}(k) \quad (2.11)$$

Berdasarkan Teorema 2.2, maka Persamaan (2.11) dapat ditulis sebagai

$$\mathbf{x}(k) = A_0^* \otimes \mathbf{b}(k)$$

atau secara eksplisit ditulis sebagai berikut,

$$\mathbf{x}(k) = A_0^* \otimes A_1 \otimes \mathbf{x}(k-1) \oplus \dots \oplus A_0^* \otimes A_M \otimes \mathbf{x}(k-M) \quad (2.12)$$

Perbedaan antara Persamaan (2.8) dengan (2.12) adalah pada Persamaan (2.12) variable keadaan $\mathbf{x}(k)$ hanya berada di sisi kiri tanda sama dengan (“=”) di dalam persamaan.

Langkah terakhir yaitu, mentransformasikan Persamaan (2.12) ke dalam bentuk persamaan relasi rekurensi orde-1, yaitu dengan terlebih dahulu memisalkan

$$\tilde{\mathbf{x}}(k) = [\mathbf{x}'(k-1), \mathbf{x}'(k-2), \dots, \mathbf{x}'(k-M)]'$$

dan

$$\tilde{A} = \begin{bmatrix} A_0^* \otimes A_1 & A_0^* \otimes A_2 & \dots & \dots & A_0^* \otimes A_M \\ E & \mathcal{E} & \dots & \dots & \mathcal{E} \\ \mathcal{E} & E & \ddots & \dots & \mathcal{E} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathcal{E} & \mathcal{E} & \dots & E & \mathcal{E} \end{bmatrix}$$

Sehingga didapatkan persamaan relasi rekurensi orde-1 adalah sebagai berikut,

$$\tilde{\mathbf{x}}(k+1) = \tilde{A}(k) \otimes \tilde{\mathbf{x}}(k), k \geq 0 \quad (2.13)$$

BAB 3

METODA PENELITIAN

Tahapan penelitian yang digunakan dalam menyelesaikan permasalahan ini adalah:

1. Studi literatur

Pada tahap ini dilakukan identifikasi permasalahan dengan mencari referensi yang menunjang penelitian. Referensi yang dipakai adalah buku, buku elektronik, jurnal ilmiah, maupun artikel dari internet.

2. Mengumpulkan dan mengolah data

Pada tahap ini dilakukan pengumpulan dan pengolahan data yang diperlukan dalam mengerjakan penelitian. Data diambil dari PT Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya yaitu data jalur dan jadwal keberangkatan kereta api Jalur Waru-Sidoarjo. Data jalur digunakan sebagai landasan membentuk model Petri Net, sedangkan jadwal keberangkatan kereta api digunakan sebagai inputan nilai parameter dalam melakukan perhitungan nilai eigen dari model Aljabar Max-Plus.

3. Mengkontruksi model Petri Net dari jalur kereta api

Pada tahap ini dikonstruksi model Petri Net dari jalur kereta api *semi-double track* yang sebelumnya diberikan struktur hirarkis arah jalur lintasannya. Pada tahap ini juga diberikan gambaran simulasi model Petri Net menggunakan alat bantu Pipe 3.

4. Mengkontruksi model Aljabar Max-Plus dari model Petri Net

Pada tahap ini dikonstruksi model Aljabar Max-Plus berdasarkan model Petri Net yang telah didapatkan pada Tahap 3.

5. Menganalisis hasil perhitungan penjadwalan model Aljabar Max-Plus

Pada tahap ini dilakukan perhitungan nilai eigen dari model Aljabar Max-Plus yang didapatkan pada Tahap 4. Dari hasil perhitungan yang dilakukan dapat diketahui sifat keperiodikan dari sistem jalur kereta api *semi-double track* yang dibuat. Dalam melakukan perhitungan ini digunakan alat bantu Scilab 5.5.0.

6. Menyusun hasil penelitian

Pada tahap ini dilakukan penulisan laporan hasil penelitian yang dilakukan mulai dari tahap studi literatur, pengumpulan dan pengolahan data, penyusunan model Petri Net dan model Aljabar Max-Plus, analisis hasil penjadwalan dari model Aljabar Max Plus.

BAB 4

HASIL DAN PEMBAHASAN

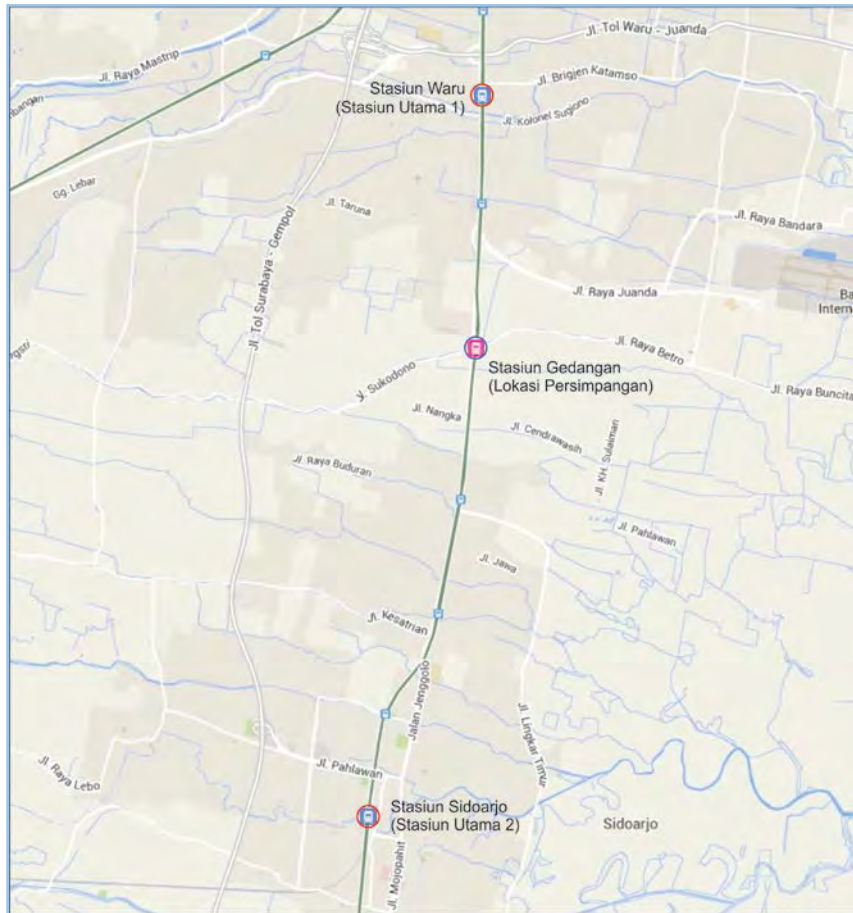
Pada bab ini diberikan hasil penelitian, yaitu analisis permasalahan, model Petri Net, dan model Aljabar Max-Plus. Adapun analisis permasalahan yang dibahas merupakan penjelasan mengenai proses pembentukan struktur hirarkis jalur kereta api. Kemudian struktur hirarkis tersebut diterapkan pada model Petri Net, selain itu diberikan simulasi dari model Petri Net dengan menggunakan alat bantu *software* Pipe 3. Dan yang terakhir, dari model Petri Net yang didapatkan diterjemahkan ke dalam model Aljabar Max-Plus, dan menghitung nilai eigen dengan menggunakan alat bantu *software* Scilab 5.5.0.

4.1 Analisis Permasalahan

Penelitian ini dilakukan pada jalur kereta api Waru-Sidoarjo (Gambar 4.1), yang terdiri dari 3 stasiun, 2 stasiun utama yaitu Stasiun Waru (WR) dan Stasiun Sidoarjo (SDA), dan 1 stasiun yang berada di antara jalur *single track* Waru-Sidoarjo yaitu Stasiun Gedangan (GDG), dalam hal ini yang difungsikan sebagai stasiun lokasi persimpangan. Sebagai tambahan ilustrasi, jalur kereta api Waru-Sidoarjo ini selanjutnya menuju ke Stasiun Wonokromo (WO) ke arah utara, dan menuju ke Stasiun Bangil (BG) ke arah selatan, satu lagi ke arah barat menuju ke Stasiun Mojokerto (MR).

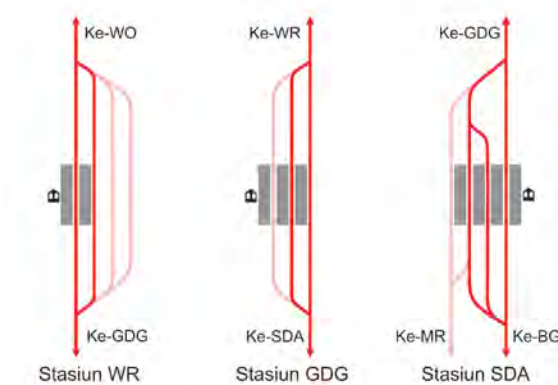
Stasiun Waru memiliki 4 jalur, dengan jalur 1 dan 2 adalah jalur untuk kereta reguler yang berhenti maupun langsung serta untuk pertemuan kereta api yang berjalan berlawanan arah, sementara 2 jalur lainnya diperuntukkan untuk kereta api peti kemas. Kereta api peti kemas ini hanya beroperasi dua hari sekali. Sehingga dalam penelitian ini disimpulkan bahwa Stasiun Waru hanya memiliki 2 jalur untuk kereta api reguler.

Sedangkan Stasiun Sidoarjo memiliki 5 jalur, akan tetapi 1 jalur sekarang sudah digusur, kemudian 1 jalur tidak beroperasi yaitu jalur yang menuju ke Stasiun Mojokerto (MR). Sehingga dalam penelitian ini disimpulkan bahwa Stasiun Sidoarjo hanya memiliki 3 jalur untuk kereta api reguler.



Gambar 4.1 Peta Jalur Kereta Api Waru-Sidoarjo

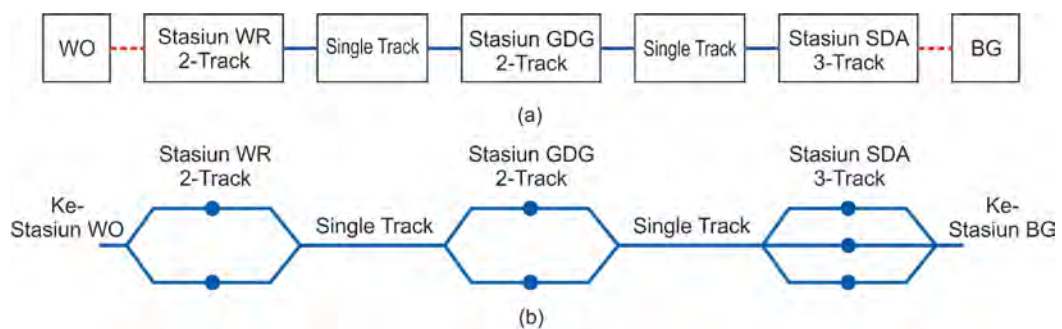
Sedangkan untuk Stasiun Gedangan pada awalnya mempunyai 3 jalur rel kereta api, namun 1 jalur dibongkar dan kini tinggal 2 jalur saja. Sehingga dalam penelitian ini disimpulkan bahwa Stasiun Gedangan hanya memiliki 2 jalur untuk kereta api reguler.



Gambar 4.2 Kondisi Stasiun Waru, Stasiun Gedangan, dan Stasiun Sidoarjo

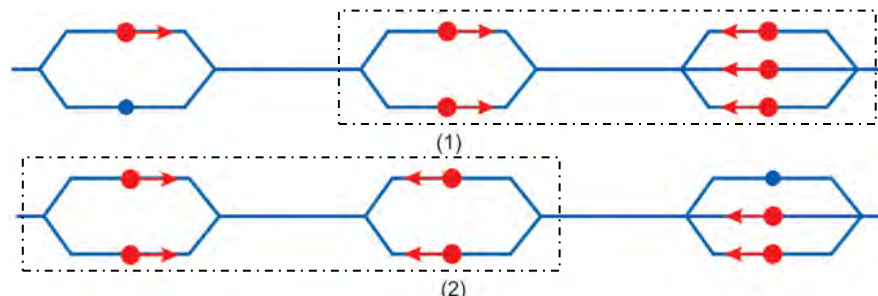
Selain ketiga stasiun tersebut terdapat 4 halte yang beroperasi pada jalur kereta api Waru-Sidoarjo, 1 halte terdapat di antara jalur Waru-Gedangan yaitu Halte Sawotratap (STP), dan 3 halte terdapat di antara jalur Gedangan-Sidoarjo yaitu Halte Banjarkemantren (BKN), Halte Buduran (BDU), dan Halte Pagerwojo (PWO). Keempat halte tersebut hanya memiliki 1 jalur, sehingga tidak diperlukan dalam penelitian ini.

Berdasarkan penjelasan kondisi jalur kereta api Waru-Sidoarjo, dapat digambarkan sebagai berikut,



Gambar 4.3 Gambaran Umum Jalur Kereta Api Waru-Sidoarjo

Setiap segmen jalur pada rel kereta api hanya dapat digunakan oleh satu kereta api saja pada saat ke- k . Titik biru pada gambar merupakan titik tempat kereta api berhenti. Untuk setiap jalur di stasiun, kereta api dapat melewatinya 2 arah, tetap dengan syarat jalur di stasiun tersebut tidak sedang dipakai untuk berhenti oleh kereta api lainnya. Keadaan yang harus dapat dihindari adalah ketika kereta api memenuhi jalur di dua Stasiun yang saling berdekatan dan dengan arah yang saling behadap-hadapan. Untuk lebih jelasnya berikut diberikan ilustrasinya.

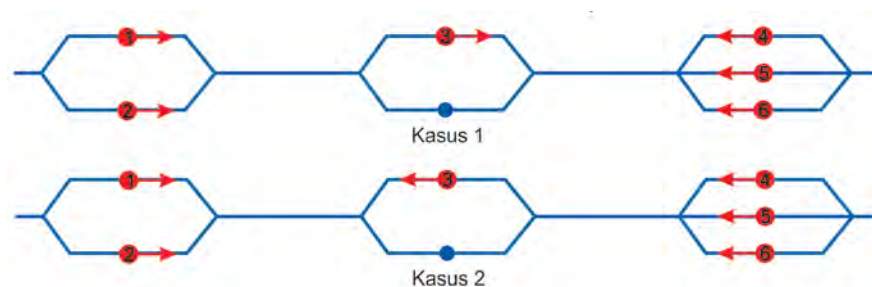


Gambar 4.4 Ilustrasi Keadaan yang Harus Dihindari

Pada Gambar 4.4 (1) mengilustrasikan bahwa saat ke- k terdapat 2 buah kereta api di GDG dan 3 buah kereta api di SDA dengan arah yang saling berhadap-hadapan. Keadaan ini menggambarkan kejadian dimana kereta api pada masing-masing stasiun tersebut tidak dapat melakukan perjalanan berikutnya, dikarenakan tidak ada jalur yang dapat dilewati saat ke- $(k + 1)$, atau dengan kata lain kereta api dari GDG tidak dapat melanjutkan perjalanan ke SDA, begitu juga sebaliknya.

Sedangkan pada Gambar 4.4 (2) mengilustrasikan bahwa saat ke- k terdapat 2 buah kereta api di WR dan 2 buah kereta api di GDG dengan arah yang saling berhadap-hadapan. Keadaan ini menggambarkan kejadian dimana kereta api pada masing-masing stasiun tersebut tidak dapat melakukan perjalanan berikutnya, dikarenakan tidak ada jalur yang dapat dilewati saat ke- $(k + 1)$, atau dengan kata lain kereta api dari WR tidak dapat melanjutkan perjalanan ke GDG, begitu juga sebaliknya.

Berikut diberikan 2 permasalahan dan bagaimana urutan penyelesaian permasalahan tersebut, untuk menghindari keadaan yang harus dihindari. Untuk memudahkan dalam menjelaskan urutan kereta api mana saja yang harus diprioritaskan, maka setiap kereta diberi label nomor. Jadi nomor tersebut hanya sebagai alat bantu dalam menjelaskan permasalahan, tidak menunjukkan sebuah arti kusus dalam istilah perkereta-apian, lihat Gambar 4.5.



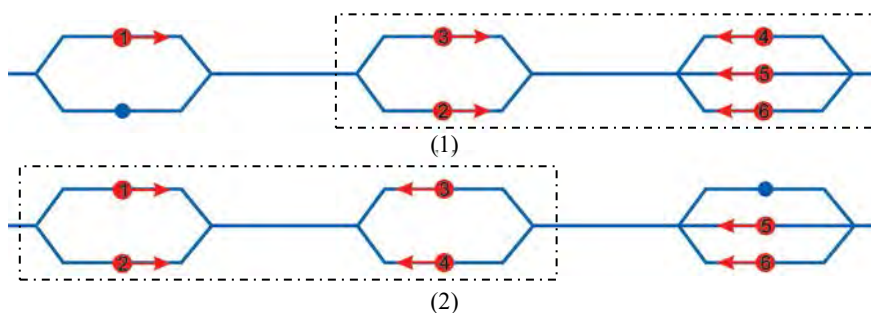
Gambar 4.5 Contoh Kasus yang Memerlukan Prioritas

Kedua kasus ini merupakan kejadian terburuk yang dapat terjadi dikarenakan lamanya durasi yang dibutuhkan oleh kereta api untuk berjalan melintasi suatu jalur single track. Sehingga mengakibatkan kereta api dari arah

berlawanan maupun kereta api yang berjalan searah harus saling menunggu kereta api yang berjalan sebelumnya hingga sampai stasiun.

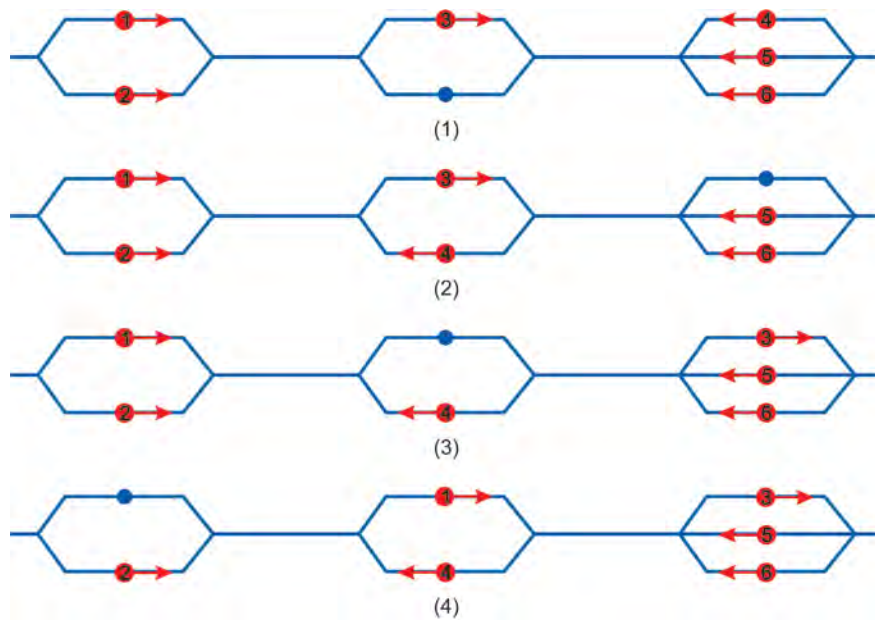
Pada permasalahan yang pertama Gambar 4.5 (Kasus 1), pada saat ke- k terdapat dua kereta api di WR yang akan menuju ke GDG, satu kereta api di GDG yang akan menuju ke SDA, dan 3 kereta api di SDA yang akan menuju ke GDG. Sedangkan pada permasalahan yang kedua Gambar 4.5 (Kasus 2), pada saat ke- k terdapat dua kereta api di WR yang akan menuju ke GDG, satu kereta api di GDG yang akan menuju ke WR, dan 3 kereta api di SDA yang akan menuju ke GDG.

Kedua permasalahan tersebut merupakan keadaan kritis, dimana urutan kereta api yang didahulukan untuk berjalan saat ke- $(k + 1)$ dapat menyebabkan *bottleneck* (keadaan sistem dimana kereta api tidak bisa berjalan untuk melanjutkan perjalanan selanjutnya), misalnya saja pada Kasus 1 jika saat ke- $(k + 1)$ kereta api pada WR didahulukan berjalan, dan pada Kasus 2 jika saat ke- $(k + 1)$ kereta api pada SDA didahulukan berjalan.



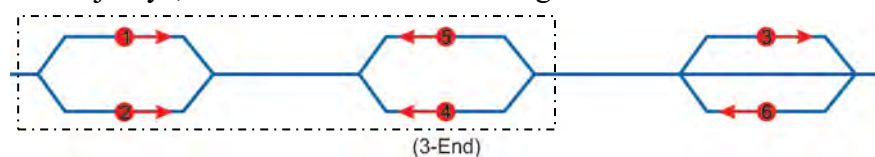
Gambar 4.6 Keadaan Sistem Jika Urutan Penyelesaian Tidak Ada Aturan Prioritas

Berikut diberikan urutan langkah-perlangkah yang dapat dilakukan dalam menyelesaikan permasalahan keadaan kritis pada Gambar 4.5. Pada Kasus 1, dimisalkan keadaan awal merupakan keadaan saat ke- k , maka kereta api yang diprioritaskan untuk didahulukan berjalan saat ke- $(k + 1)$ adalah kereta api yang berada di SDA, lihat Gambar 4.7 (2). Dengan begitu memberikan akses jalan bagi kereta api yang berada di GDG untuk berjalan saat ke- $(k + 2)$ menuju ke arah SDA, lihat Gambar 4.7 (3). Berbeda keadaannya jika kereta api yang berada di WR didahulukan untuk berjalan, ini akan menyebabkan *bottleneck* seperti pada Gambar 4.6 (1).



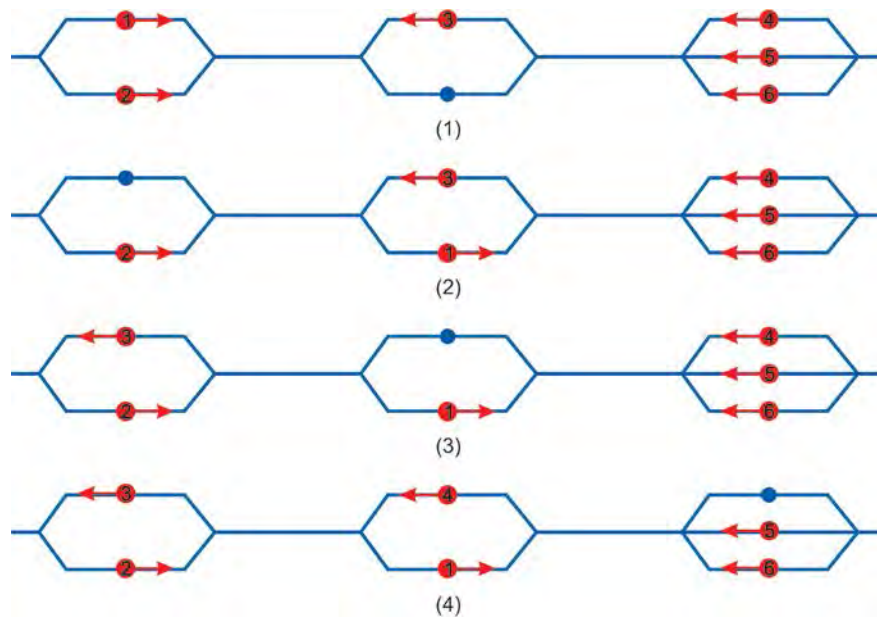
Gambar 4.7 Urutan Prioritas Penyelesaian Kasus 1

Kejadian saat ke- $(k + 2)$ ini menyebabkan keadaan kritis berikutnya. Untuk kejadian saat ke- $(k + 3)$ kita harus mendahulukan kereta api dari WR menuju ke GDG, sehingga memberikan akses jalan pada kereta api yang berada di GDG untuk berjalan saat ke- $(k + 4)$ ke arah WR. Jika saat ke- $(k + 3)$ kereta api dari SDA menuju ke GDG didahulukan berjalan, maka akan terjadi keadaan *bottleneck* lainnya, seperti yang terlihat pada Gambar 4.8 berikut ini. Untuk keadaan selanjutnya, sudah tidak ada masalah lagi.



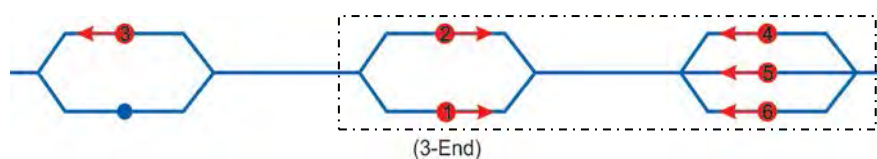
Gambar 4.8 Kejadian *Bottleneck* Lainnya pada Kasus 1

Sedangkan pada Kasus 2, dimisalkan keadaan awal merupakan keadaan saat ke- k , maka kereta api yang diprioritaskan untuk didahulukan berjalan saat ke- $(k + 1)$ adalah kereta api yang berada di WR, lihat Gambar 4.9 (2). Dengan begitu memberikan akses jalan bagi kereta api yang berada di GDG untuk berjalan saat ke- $(k + 2)$ menuju ke arah WR, lihat Gambar 4.9 (3). Berbeda keadaannya jika kereta api yang berada di SDA didahulukan berjalan, ini akan menyebabkan *bottleneck* seperti pada Gambar 4.6 (2).



Gambar 4.9 Urutan Prioritas Penyelesaian Kasus 2

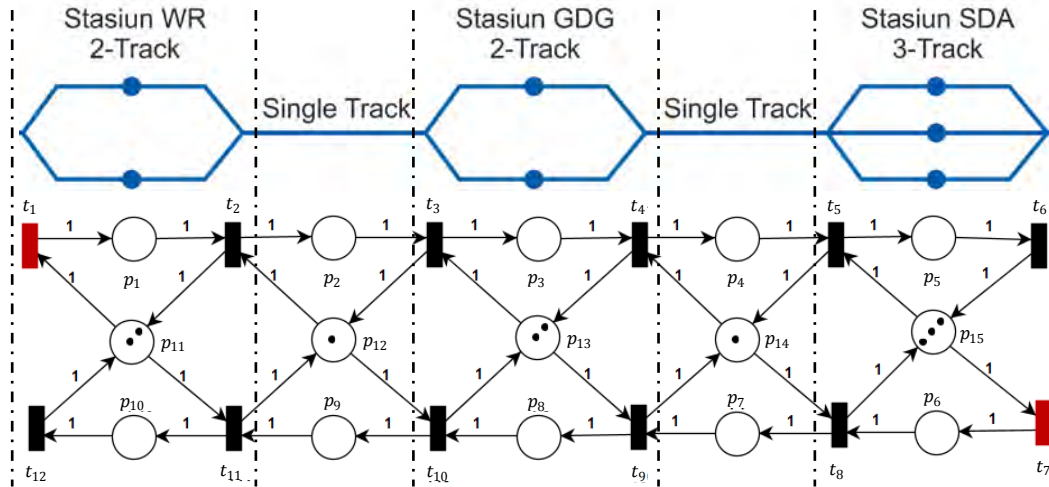
Kejadian saat $ke-(k + 2)$ ini menyebabkan keadaan kritis berikutnya. Untuk kejadian saat $ke-(k + 3)$ kita harus mendahulukan kereta api dari SDA menuju ke GDG, sehingga memberikan akses jalan pada kereta api yang berada di GDG untuk berjalan saat $ke-(k + 4)$ menuju ke arah SDA. Jika saat $ke-(k + 3)$ kereta api dari WR menuju ke GDG didahulukan berjalan, maka akan terjadi keadaan *bottleneck* lainnya seperti yang terlihat pada Gambar 4.10 berikut ini. Untuk keadaan selanjutnya, sudah tidak ada masalah lagi.



Gambar 4.10 Kejadian *Bottleneck* Lainnya pada Kasus 2

4.2 Model Petri Net

Berdasarkan kondisi jalur kereta api Waru-Sidoarjo dan beberapa penjelasan yang sudah dipaparkan pada Sub-Bab 4.1, berikut disusun suatu model Petri Net jaringan kereta api jalur Waru-Sidoarjo seperti yang terlihat pada Gambar 4.11. Model Petri Net tersebut merupakan model Petri Net sebelum mempertimbangkan urutan prioritas.

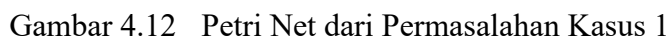


Gambar 4.11 Model Petri Net dari Jalur Kereta Api Waru-Sidoarjo

Dalam hal ini P merupakan himpunan berhingga *place* yaitu $P = \{p_1, p_2, p_3, p_4, p_5, \dots, p_{15}\}$ dan jumlah *token* yang terdapat pada masing-masing *place* $p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}$ menunjukkan keberadaan kereta api saat ke- k , $p_{11}, p_{12}, p_{13}, p_{14}, p_{15}$ menunjukkan jalur yang dapat digunakan kereta api untuk melintas ke stasiun berikutnya saat ke- k . Sedangkan T merupakan himpunan berhingga *transition*, $T = \{t_1, t_2, t_3, t_4, \dots, t_{12}\}$. *Transition* menunjukkan *event* keberangkatan atau kedatangan kereta api di tiap-tiap stasiun.

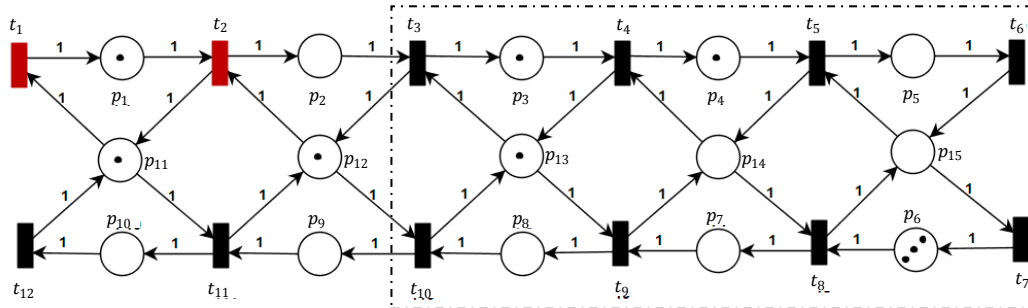
Pada Gambar 4.11 merupakan Petri Net awal yang dikonstruksi tanpa mempertimbangkan urutan prioritas. Petri Net tersebut merupakan keadaan awal dimana masih belum ada kereta api di ketiga stasiun maupun di jalur yang menghubungkan tiap-tiap stasiun. Warna merah pada *transition* merupakan penanda bahwa *transition* tersebut *enabled*, yang artinya *event* terpenuhi, dalam hal ini kereta api dapat melakukan perjalanan berikutnya. Sedangkan arti dari jumlah *token* pada masing-masing *place* adalah sebagai berikut: Pertama, pada *place* p_{11} terdapat 2 *token* yang berarti bahwa ada 2 jalur di WR yang sedang tidak digunakan oleh kereta api untuk berhenti. Kedua, pada *place* p_{12} terdapat 1 *token* yang berarti bahwa jalur yang menghubungkan WR dengan GDG sedang tidak digunakan oleh kereta api untuk berjalan. Ketiga, pada *place* p_{13} terdapat 2 *token* yang berarti bahwa ada 2 jalur di GDG yang sedang tidak digunakan oleh kereta api untuk berhenti. Keempat, pada *place* p_{14} terdapat 1 *token* yang berarti

Selain itu Petri Net pada Gambar 4.11 tidak memperhatikan urutan prioritas. Semua kereta api dapat berjalan pada masing-masing segmen jalur hanya dengan melihat bahwa keadaan segmen jalur di depannya minimal satu saja dalam keadaan sedang tidak digunakan (*idle*).



Berdasarkan gambar Gambar 4.12 terlihat bahwa ada 3 *transition* berwarna merah (*Transition* t_2 , t_4 , dan t_8), artinya ada 3 kereta api yang dapat berangkat, yaitu kereta api dari WR menuju ke WR-GDG, kereta api dari GDG menuju ke GDG-SDA, dan kereta api dari SDA menuju ke SDA-GDG. Dalam hal ini urutan jalannya kereta api akan mempengaruhi keadaan selanjutnya, misalkan jika kereta api dari WR menuju ke GDG diberangkatkan lebih dahulu (*transition* t_2 di-*fire*), maka akan terjadi *deadlock* di SDA atau di GDG karena tidak ada jalur yang dalam keadaan *idle* di SDA untuk tempat melintas kereta api dari GDG, begitu juga sebaliknya, tidak ada jalur yang dalam keadaan *idle* di GDG untuk tempat melintas kereta api dari SDA. Hal ini sesuai dengan keadaan *bottleneck* pada kasus 1 pada ilustrasi Gambar 4.6 (1), bandingkan gambar tersebut dengan Gambar 4.13, gambar tersebut mengilustrasikan jika kemudian kereta api dari

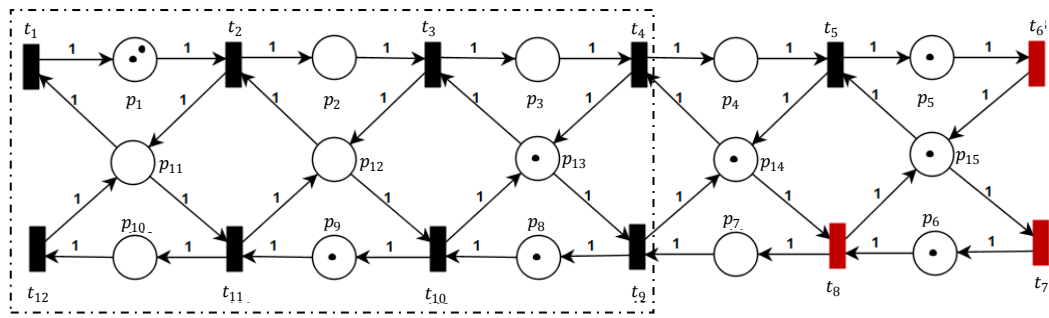
GDG menuju ke SDA diberangkatkan (*transition t_4 di-fire*), maka akan terjadi *deadlock*. Sama halnya jika kereta api dari SDA menuju ke GDG diberangkatkan (*transition t_8 di-fire*), maka akan terjadi *deadlock* juga.



Gambar 4.13 Ilustrasi Kejadian *Deadlock* dari Kasus 1

Begitu juga jika kereta api dari SDA menuju ke GDG-SDA diberangkatkan dan berjalan hingga ke GDG (*transition t_8 di-fire*, kemudian *transition t_9*), dilanjutkan dengan kereta api dari GDG menuju ke GDG-SDA dan berjalan hingga ke SDA (*transition t_4 di-fire*, kemudian *t_5 di-fire*), dan dilanjutkan dengan kereta api dari SDA menuju ke GDG-SDA diberangkatkan dan berjalan hingga ke GDG (*transition t_8 di-fire*, kemudian *transition t_9*), maka akan terjadi *deadlock* di WR atau di GDG karena tidak ada jalur yang dalam keadaan *idle* di WR untuk tempat melintas kereta api dari GDG, begitu juga sebaliknya, tidak ada jalur yang dalam keadaan *idle* di GDG untuk tempat melintas kereta api dari WR. Hal ini sesuai dengan keadaan *bottleneck* lainnya pada kasus 1 pada ilustrasi Gambar 4.8, bandingkan gambar tersebut dengan Gambar 4.14, gambar tersebut mengilustrasikan jika kemudian kereta api dari GDG menuju ke WR diberangkatkan (*transition t_{10} di-fire*), maka akan terjadi *deadlock*. Sama halnya jika kereta api dari WR menuju ke GDG diberangkatkan (*transition t_2 di-fire*), maka akan terjadi *deadlock* juga.

Kejadian *deadlock* ini masih banyak ditemui pada lintasan alur selanjutnya. Hal ini juga terjadi pada kasus 2 yang diberikan jika diimplementasikan pada model Petri Net di atas (Gambar 4.11) maka terdapat kejadian yang menyebabkan terjadinya *deadlock*, sehingga model Petri Net tersebut masih perlu perbaikan dengan cara mempertimbangkan urutan prioritas.



Gambar 4.14 Ilustrasi Kejadian *Deadlock* Lainnya dari Kasus 1

Berikut diberikan model Petri Net setelah mempertimbangkan urutan prioritas ke dalam model yang dikonstruksi (Gambar 4.15), sehingga membentuk suatu struktur hirarkis arah jalur lintasan kereta api. Model Petri Net ini dikonstruksi berdasarkan observasi dari permasalahan-permasalahan yang telah dijelaskan pada Sub-Bab 4.1, dengan tujuan menghindari terjadinya *deadlock*.

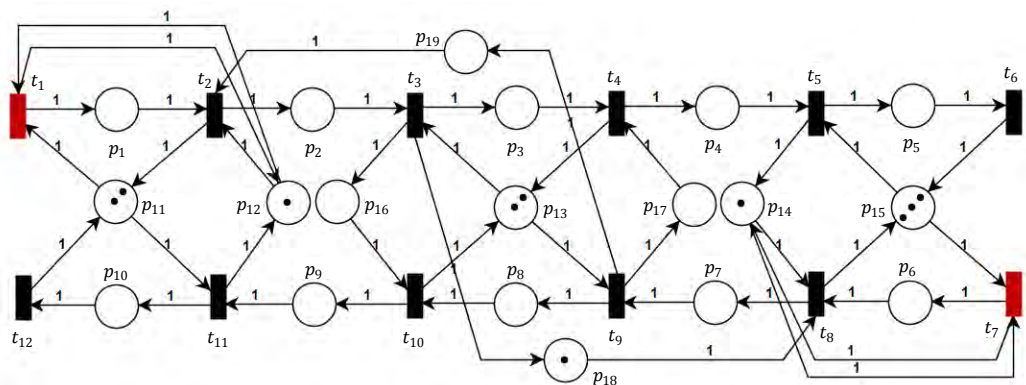
Sebagai penjelasan ilustrasi, untuk kedatangan kereta api dari WO menuju ke WR harus melihat beberapa hal, pertama sudah jelas kedatangan kereta api dari WO menuju ke WR ini harus terlebih dahulu melihat bahwa ada jalur di WR minimal satu dalam keadaan *idle*. Kedua, kedatangan kereta api dari WO menuju ke WR harus terlebih dahulu melihat bahwa pada jalur yang menghubungkan WR dengan GDG harus dalam keadaan *idle*, ini dimaksudkan untuk menghindari keadaan dimana jalur di WR tersebut hanya tinggal satu, dan jalur satunya sudah digunakan kereta api dari WO yang berangkat lebih dahulu sebelumnya, bersamaan dengan itu ada kereta api yang berjalan berlawanan arah dari GDG menuju ke WR. Sehingga kereta api ini harus diprioritaskan untuk tiba di WR terlebih dahulu dibandingkan dengan kereta api datang dari WO.

Untuk keberangkatan kereta api dari WR menuju ke GDG harus menunggu jalur yang menghubungkan antara WR dengan GDG dalam keadaan *idle*. Selain itu harus melihat prioritas yang diberikan yaitu prioritas mendahulukan kereta api dari WR berjalan menuju ke GDG atau mendahulukan kereta api dari SDA berjalan menuju ke GDG terlebih dahulu. Hal ini dalam model Petri Net Gambar 4.15 ditandai dengan *place* yang menjadi tambahan pada model Petri Net Gambar 4.14 (*place* p_{18} dan *place* p_{19}).

Sedangkan untuk keberangkatan kereta api dari GDG menuju ke SDA hanya perlu memperhatikan bahwa jalur yang menghubungkan antara GDG dengan SDA dalam keadaan *idle*. Keberangkatan kereta api dari GDG menuju ke SDA tidak perlu melihat ada atau tidaknya jalur di SDA minimal satu dalam keadaan *idle*. Hal ini dikarenakan sudah dijamin pasti minimal ada satu jalur dalam keadaan *idle*. Dalam model Petri Net Gambar 4.15 ditandai dengan *place* yang menjadi tambahan dari model Petri Net Gambar 4.14 (*place p₁₆*).

Hal-hal yang dijelaskan di atas berlaku serupa dengan kereta api yang berjalan dari arah yang berlawanan, misalnya kedatangan kereta api dari BG menuju ke SDA serupa dengan kedatangan kereta api dari WO menuju ke WR, keberangkatan kereta api dari WR menuju ke GDG serupa dengan keberangkatan kereta api dari SDA menuju ke GDG, dan keberangkatan kereta api dari GDG menuju ke WR serupa dengan keberangkatan kereta api dari GDG menuju ke SDA (dengan tambahan *place p₁₇*). Sedangkan untuk kedatangan dan keberangkatan kereta api yang lainnya hanya melihat bahwa ada atau tidaknya jalur yang berada di depannya minimal satu dalam keadaan *idle*.

Berikut hasil kontruksi model Petri Net yang didapatkan dengan mempertimbangkan urutan prioritas berdasarkan beberapa penjelasan di atas (lihat Gambar 4.15). Model Petri Net ini tidak jauh berbeda dengan model Petri Net tanpa prioritas pada Gambar 4.11, hanya saja ada penambahan empat buah *place* dan beberapa *arc*, serta adanya perubahan beberapa *arc* yang berfungsi sebagai pengatur struktur hirarkis arah jalur lintasan kereta api.



Gambar 4.15 Model Petri Net dengan Prioritas

Jika permasalahan yang diberikan di Sub-Bab 4.1 pada Gambar 4.5 diterapkan pada model Petri Net pada Gambar 4.15, maka kejadian-kejadian selanjutnya akan diarahkan sehingga tidak akan ada terjadi keadaan *deadlock*. untuk selanjutnya model Petri Net ini akan diterjemahkan ke dalam model Aljabar Max-Plus pada bahasan Sub-Bab 4.3.

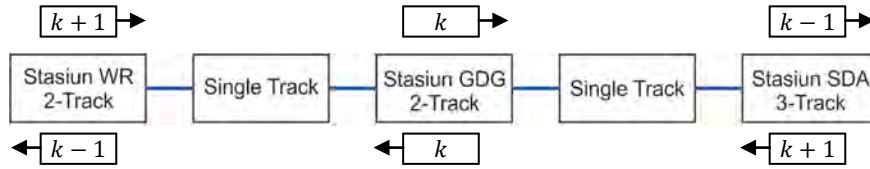
4.3 Model Aljabar Max-Plus

Pada model Petri Net yang telah didapatkan pada Gambar 4.15, terdapat dua *transition* input yaitu *transition* yang menandakan kedatangan kereta api dari arah WO menuju ke WR (*Transition* t_1) dan *transition* yang menandakan kedatangan kereta api dari arah BG menuju ke SDA (*Transition* t_7). Selain itu model Petri Net tersebut mempunyai dua *transition* output, yaitu *transition* yang menandakan keberangkatan kereta api dari arah SDA menuju ke BG (*Transition* t_6) dan *transition* yang menandakan keberangkatan kereta api dari arah WR menuju ke WO (*Transition* t_{12}).

Dalam melakukan pemodelan Aljabar Max-Plus, pada model Petri Net Gambar 4.15 ini dianalogikan terdapat 2 *place* yang menghubungkan *transition* input dan *transition* output, yaitu *transition* t_6 dengan *transition* t_1 dan *transition* t_{12} dengan *transition* t_7 . *Place* ini difungsikan sebagai penanda selang waktu antara kereta api yang keluar dari sistem dengan kereta api yang masuk ke dalam sistem. Dalam hal ini, jika ada n kereta api pada saat ke- k yang berjalan dari arah WR menuju ke SDA, maka untuk kereta api yang datang dari arah WO menuju ke WR berikutnya $(k + 1)$ harus menunggu kereta api ke- $(k + 1 - n)$ di SDA berjalan menuju ke BG dalam selang waktu tertentu.

Untuk mempermudah dalam menjelaskan pembentukan model Aljabar Max-Plus berikut diberikan ilustrasi posisi saat terjadinya pertemuan antara kereta api yang berjalan dari arah WR menuju ke SDA dengan kereta api yang berjalan dari arah SDA menuju ke WR. Seperti pada penjelasan-penjelasan sebelumnya, diketahui bahwa setiap kereta api yang berajalan dari WR menuju ke SDA akan saling berbagi *resource* dengan kereta api yang berjalan dari arah SDA menuju ke WR atau sebaliknya. Berdasarkan Gambar 4.16, diketahui bahwa

terdapat 3 lokasi pertemuan kereta api yang dapat dilakukan yaitu di WR, GDG, dan SDA.



Gambar 4.16 Ilustrasi Posisi Saat Terjadinya Pertemuan Kereta Api yang Berjalan Berlawanan Arah

Perhatikan Gambar 4.16, jika dimisalkan pertemuan kereta api yang berjalan dari arah WR menuju ke SDA saat ke- k dengan kereta api yang berjalan dari arah SDA menuju ke WR saat ke- k terjadi di GDG, maka pertemuan kereta api yang berjalan dari arah WR menuju ke SDA saat ke- $(k + 1)$ akan terjadi di WR dengan kereta api yang berjalan dari arah SDA menuju ke WR saat ke- $(k - 1)$. Selain itu pertemuan kereta api yang berjalan dari arah WR menuju ke SDA saat ke- $(k - 1)$ akan terjadi di SDA dengan kereta api yang berjalan dari arah SDA menuju ke WR saat ke- $(k + 1)$.

Untuk keperluan selanjutnya diberikan pendefinisian variabel-variabel, yaitu variabel yang menunjukkan waktu dan variabel yang menunjukkan lama waktu, berikut variabel yang menunjukkan waktu.

- $t_1(k)$: waktu kedatangan kereta api dari arah WO (Luar Sistem) ke WR pada saat ke- k
- $t_2(k)$: waktu keberangkatan kereta api dari arah WR ke GDG pada saat ke- k
- $t_3(k)$: waktu kedatangan kereta api dari arah WR ke GDG pada saat ke- k
- $t_4(k)$: waktu keberangkatan kereta api dari GDG ke SDA pada saat ke- k
- $t_5(k)$: waktu kedatangan kereta api dari arah GDG ke SDA pada saat ke- k
- $t_6(k)$: waktu keberangkatan kereta api dari arah SDA ke BG (Luar Sistem) pada saat ke- k
- $t_7(k)$: waktu kedatangan kereta api dari arah BG (Luar Sistem) menuju ke SDA saat ke- k
- $t_8(k)$: waktu keberangkatan kereta api dari arah SDA menuju ke GDG pada saat ke- k

- $t_9(k)$: waktu kedatangan kereta api dari arah SDA ke GDG pada saat ke- k
 $t_{10}(k)$: waktu keberangkatan kereta api dari arah GDG ke WR pada saat ke- k
 $t_{11}(k)$: waktu kedatangan kereta api dari arah GDG ke WR pada saat ke- k
 $t_{12}(k)$: waktu keberangkatan kereta api dari arah WR ke WO (Luar Sistem) pada saat ke- k

Adapun variabel yang menunjukkan lama waktu, yaitu

- a_1 : lama kedatangan kereta api dari WO (Luar Sistem) di WR saat ke- k
 $a_{2,1}$: lama kereta api berhenti di WR saat ke- k
 $a_{3,2}$: lama kereta api berjalan dari arah WR menuju GDG saat ke- k
 $a_{4,3}$: lama kereta api berhenti di GDG saat ke- k
 $a_{5,4}$: lama kereta api berjalan dari arah GDG menuju SDA saat ke- k
 $a_{6,5}$: lama kereta api berhenti di SDA saat ke- k
 a_7 : lama kedatangan kereta api dari BG (Luar Sistem) di SDA saat ke- k
 $a_{8,7}$: lama kereta api berhenti di SDA saat ke- k
 $a_{9,8}$: lama kereta api berjalan dari arah SDA menuju GDG saat ke- k
 $a_{10,9}$: lama kereta api berhenti di GDG saat ke- k
 $a_{11,10}$: lama kereta api berjalan dari arah GDG menuju WR saat ke- k
 $a_{12,11}$: lama kereta api berhenti di WR saat ke- k
 b : lama waktu tunggu kereta api untuk berjalan setelah kedatangan kereta api dari arah berlawanan.

Berdasarkan model Petri Net yang sudah dikontruksi pada Sub-Bab 4.2 Gambar 4.15 dan beberapa penjelasan tambahan di atas, maka dapat dibentuk model Aljabar Max-Plus untuk perjalanan kereta api dari WR menuju ke SDA sebagai berikut.

Transition t_1 merupakan *transition* yang menandakan kedatangan kereta api berikutnya ($k + 1$) dari WO (dari luar sistem) tiba di WR. Kedatangan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun yang berjalan berlawanan arah di WR. Pertama, kereta api yang datang ke WR tersebut harus menunggu kedatangan kereta api yang berangkat dari WO (luar sistem). Dengan kata lain adalah menunggu kereta api ($k - 2$) dari SDA menuju ke BG (keluar sistem) dalam selang waktu yang ditentukan

(*Transition* t_6). Kedua, kereta api tersebut harus menunggu keberangkatan kereta api $(k - 1)$ dari WR menuju ke GDG (*Transition* t_2), karena terdapat maksimal 2 jalur yang dapat dipakai atau maksimal ada 2 kereta api di WR yang siap berangkat menuju ke GDG. Jadi, secara logika harus menunggu kereta api yang berangkat paling depan atau kereta api yang ke- $(k - 1)$ dan hingga kereta api tersebut sampai di GDG. Ketiga, harus menunggu keberangkatan kereta api $(k - 2)$ dari WR menuju ke WO (keluar sistem) (*Transition* t_{12}). Secara logika, kereta api yang berjalan berlawanan arah adalah kereta api $(k - 1)$, dan karena terdapat maksimal 2 jalur yang dapat dipakai atau maksimal ada 2 kereta api di WR yang siap berangkat menuju ke WO, dengan kata lain kereta api $(k - 2)$. Keempat, harus menunggu kedatangan kereta api (k) dari GDG menuju ke WR (*Transition* t_{11}). Secara matematis dapat ditulis sebagai berikut,

$$t_1(k + 1) = [a_1 \otimes t_6(k - 2)] \oplus [b \otimes t_2(k - 1)] \oplus [b \otimes t_{12}(k - 2)] \oplus [b \otimes t_{11}(k)] \quad (4.1)$$

Transition t_2 merupakan *transition* yang menandakan keberangkatan kereta api yang berikutnya $(k + 1)$ dari WR menuju ke GDG. Keberangkatan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun yang berjalan berlawanan arah di WR. Pertama, kereta api tersebut harus menunggu kedatangan kereta api $(k + 1)$ di WR (*Transition* t_1). Kedua, harus menunggu kereta api yang berjalan (k) dari GDG tiba di WR (*Transition* t_{11}). Ketiga, harus menunggu kereta api yang berjalan $(k + 1)$ dari SDA tiba di GDG (*Transition* t_9). Secara matematis dapat ditulis sebagai berikut,

$$t_2(k + 1) = [a_{2,1} \otimes t_1(k + 1)] \oplus [b \otimes t_9(k + 1)] \oplus [b \otimes t_{11}(k)] \quad (4.2)$$

Transition t_3 merupakan *Transition* yang menandakan kedatangan kereta api yang berikutnya $(k + 1)$ dari WR di GDG. Kedatangan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun yang berjalan berlawanan arah di GDG. Pertama, kedatangan kereta api ini harus menunggu keberangkatan kereta api $(k + 1)$ dari WR menuju ke GDG (*Transition* t_2). Kedua, harus menunggu kereta api yang berjalan di depannya

yaitu yang ke- $(k - 1)$ berjalan menuju ke SDA (*Transition* t_4) karena terdapat maksimal 2 jalur yang dapat dipakai atau maksimal ada 2 kereta api yang siap berangkat menuju ke SDA. Secara logika, harus menunggu kereta api yang berangkat paling depan atau kereta api yang ke- $(k - 1)$. Ketiga, harus menunggu keberangkatan kereta api $(k - 1)$ dari GDG berjalan menuju ke WR (*Transition* t_{10}). Hal ini serupa dengan alasan di atas, terdapat maksimal 2 jalur yang dapat dipakai atau maksimal ada dua kereta, jadi secara logika harus menunggu kereta api yang berangkat paling depan atau kereta api yang ke- $(k - 1)$. Secara matematis dapat ditulis sebagai berikut,

$$t_3(k + 1) = [a_{3,2} \otimes t_2(k + 1)] \oplus [b \otimes t_4(k - 1)] \oplus [b \otimes t_{10}(k - 1)] \quad (4.3)$$

Transition t_4 merupakan *transition* yang menandakan keberangkatan kereta api yang berikutnya $(k + 1)$ dari GDG menuju ke SDA. Keberangkatan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun yang berjalan berlawanan arah di GDG. Pertama, keberangkatan kereta api ini harus menunggu kedatangan kereta api sebelumnya $(k + 1)$ dari WR (*Transition* t_3). Kedua, harus menunggu kedatangan kereta api $(k + 1)$ di GDG (*Transition* t_9). Secara matematis dapat ditulis sebagai berikut,

$$t_4(k + 1) = [a_{4,3} \otimes t_3(k + 1)] \oplus [b \otimes t_9(k + 1)] \quad (4.4)$$

Transition t_5 merupakan *transition* yang menandakan kedatangan kereta api yang berikutnya $(k + 1)$ dari GDG di SDA. Kedatangan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun yang berjalan berlawanan arah di SDA. Pertama, kedatangan kereta api ini harus menunggu keberangkatan kereta api sebelumnya $(k + 1)$ dari GDG menuju ke SDA (*Transition* t_4). Kedua, harus menunggu kereta api $(k - 2)$ di SDA berangkat menuju ke BG (keluar sistem) (*Transition* t_6), karena terdapat maksimal 3 jalur yang dapat dipakai atau maksimal ada 3 kereta api yang siap berangkat menuju ke BG, dengan kata lain harus menunggu kereta api yang berangkat paling depan atau kereta api yang ke- $(k - 2)$. Ketiga, harus menunggu keberangkatan kereta api yang ke- $(k - 1)$ dari SDA berjalan menuju ke GDG

(*Transition* t_8). Dari beberapa penjelasan tersebut secara matematis dapat ditulis sebagai berikut,

$$t_5(k + 1) = [a_{5,4} \otimes t_4(k + 1)] \oplus [b \otimes t_6(k - 2)] \oplus [b \otimes t_8(k - 2)] \quad (4.5)$$

Transition t_6 merupakan *transition* yang menandakan keberangkatan kereta api yang berikutnya ($k + 1$) dari SDA menuju BG (keluar sistem). Keberangkatan kereta api ini harus menunggu kedatangan kereta api sebelumnya ($k + 1$) dari GDG (*Transition* t_5). Dari beberapa penjelasan tersebut secara matematis dapat ditulis sebagai berikut,

$$t_6(k + 1) = [a_{6,5} \otimes t_5(k + 1)] \quad (4.6)$$

Begitu juga untuk perjalanan kereta api dari SDA menuju ke WR, perjalanan kereta api ini sama halnya dengan perjalanan kereta api dari WR menuju ke SDA, hanya saja dengan arah yang berlawanan. Berdasarkan model Petri Net yang sudah dikonstruksi pada Sub-Bab 4.2 Gambar 4.15, maka dapat dibentuk model Aljabar Max-Plus untuk perjalanan kereta api dari SDA menuju ke WR sebagai berikut.

Transition t_7 merupakan *transition* yang menandakan kedatangan kereta api berikutnya ($k + 1$) dari BG (dari luar sistem) tiba di SDA. Kedatangan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun yang berjalan berlawanan arah di SDA. Pertama, kereta api yang datang ke SDA tersebut harus menunggu kedatangan kereta api yang berangkat dari BG (luar sistem). Dengan kata lain adalah menunggu kereta api ($k - 2$) dari WR menuju ke WO (keluar sistem) dalam selang waktu yang ditentukan (*Transition* t_{12}). Kedua, kereta api tersebut harus menunggu keberangkatan kereta api ($k - 2$) dari SDA menuju ke GDG (*Transition* t_8), karena terdapat maksimal 3 jalur yang dapat dipakai atau maksimal ada 3 kereta api di SDA yang siap berangkat menuju ke GDG. Jadi, secara logika harus menunggu kereta api yang berangkat paling depan atau kereta yang ke- $(k - 2)$ dan hingga kereta api tersebut sampai di GDG. Ketiga, harus menunggu keberangkatan kereta api ($k - 3$) dari SDA menuju ke BG (luar sistem) (*Transition* t_6). Secara logika, kereta api yang berjalan berlawanan arah adalah

kereta api $(k - 1)$, dan karena terdapat maksimal 3 jalur yang dapat dipakai atau maksimal ada 3 kereta api di SDA yang siap berangkat menuju ke BG, dengan kata lain, kereta api $(k - 3)$. Keempat, kereta api tersebut harus menunggu kedatangan kereta api (k) dari GDG menuju ke SDA (*Transition* t_5). Dari beberapa penjelasan tersebut secara matematis dapat ditulis sebagai berikut,

$$t_7(k + 1) = [a_7 \otimes t_{12}(k - 2)] \oplus [b \otimes t_8(k - 2)] \oplus [b \otimes t_6(k - 3)] \oplus [b \otimes t_5(k)] \quad (4.7)$$

Transition t_8 merupakan *transition* yang menandakan keberangkatan kereta api yang berikutnya $(k + 1)$ dari SDA menuju ke GDG. Keberangkatan kereta api harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun kereta api yang berjalan berlawanan arah di SDA. Pertama, kereta api tersebut harus menunggu kedatangan kereta api $(k + 1)$ tiba di SDA (*Transition* t_7). Kedua, harus menunggu kereta api yang berjalan (k) dari GDG tiba di SDA (*Transition* t_5). Ketiga, harus menunggu kereta api yang berjalan (k) dari WR tiba di GDG (*Transition* t_3). Dari beberapa penjelasan tersebut secara matematis dapat ditulis sebagai berikut,

$$t_8(k + 1) = [a_{8,7} \otimes t_7(k + 1)] \oplus [b \otimes t_3(k)] \oplus [b \otimes t_5(k)] \quad (4.8)$$

Transition t_9 merupakan *transition* yang menandakan kedatangan kereta api yang berikutnya $(k + 1)$ dari SDA di GDG. Kedatangan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun yang berjalan berlawanan arah di GDG. Pertama, kedatangan kereta api ini harus menunggu keberangkatan kereta api $(k + 1)$ dari SDA menuju ke GDG (*Transition* t_8). Kedua, harus menunggu kereta api yang berjalan di depannya $(k - 1)$ berjalan menuju ke WR (*Transition* t_{10}) karena terdapat maksimal 2 jalur yang dapat dipakai atau maksimal ada 2 kereta api yang siap berangkat menuju ke WR. Secara logika, harus menunggu kereta api yang berangkat paling depan atau kereta api yang ke- $(k - 1)$. Ketigab, harus menunggu keberangkatan kereta api $(k - 1)$ dari GDG berjalan menuju ke SDA (*Transition* t_4). Hal ini serupa dengan alasan di atas, terdapat maksimal 2 jalur yang dapat dipakai atau maksimal ada dua kereta, jadi secara logika harus menunggu kereta api yang berangkat paling

depan atau kereta api yang ke- $(k - 1)$. Dari beberapa penjelasan tersebut secara matematis dapat ditulis sebagai berikut,

$$t_9(k + 1) = [a_{9,8} \otimes t_8(k + 1)] \oplus [b \otimes t_4(k - 1)] \oplus [b \otimes t_{10}(k - 1)] \quad (4.9)$$

Transition t_{10} merupakan *transition* yang menandakan keberangkatan kereta api yang berikutnya $(k + 1)$ dari GDG menuju ke WR. Keberangkatan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjela di depannya maupun yang berjalan berlawanan arah di GDG. Pertama, keberangkatan kereta api ini harus menunggu kedatangan kereta api sebelumnya $(k + 1)$ dari SDA (*Transition* t_9). Kedua, harus menunggu kedatangan kereta api $(k + 1)$ di GDG (*Transition* t_3). Dari beberapa penjelasan tersebut secara matematis dapat ditulis sebagai berikut,

$$t_{10}(k + 1) = [a_{10,9} \otimes t_9(k + 1)] \oplus [b \otimes t_3(k + 1)] \quad (4.10)$$

Transition t_{11} merupakan *transition* yang menandakan kedatangan kereta api yang berikutnya $(k + 1)$ dari GDG menuju WR. Kedatangan kereta api ini harus menunggu kedatangan dan keberangkatan kereta api yang berjalan di depannya maupun yang berjalan berlawanan arah di WR. Pertama, kedatangan kereta api ini harus menunggu keberangkatan kereta api sebelumnya $(k + 1)$ dari GDG (*Transition* t_{10}). Kedua, harus menunggu kereta api $(k - 1)$ di WR berangkat menuju ke WO (keluar sistem) (*Transition* t_{12}), karena terdapat maksimal 2 jalur yang dapat dipakai atau maksimal ada 2 kereta api yang siap berangkat menuju ke WO, dengan kata lain harus menunggu kereta api yang berangkat paling depan atau kereta api yang ke- $(k - 1)$. Ketiga, harus menunggu keberangkatan kereta api (k) dari WR berjalan menuju ke GDG (*Transition* t_2). Dari beberapa penjelasan tersebut secara matematis dapat ditulis sebagai berikut,

$$t_{11}(k + 1) = [a_{11,10} \otimes t_{10}(k + 1)] \oplus [b \otimes t_{12}(k - 1)] \oplus [b \otimes t_2(k)] \quad (4.11)$$

Transition t_{12} merupakan *transition* yang menandakan keberangkatan kereta api yang berikutnya $(k + 1)$ dari WR menuju WO (keluar sistem). Keberangkatan kereta api ini harus menunggu kedatangan kereta api sebelumnya

yaitu yang ke- $(k + 1)$ dari GDG (*Transition* t_{11}). Dari beberapa penjelasan tersebut secara matematis dapat ditulis sebagai berikut,

$$t_{12}(k + 1) = [a_{12,11} \otimes t_{11}(k + 1)] \quad (4.12)$$

Dari Persamaan (4.1), (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), dan (4.12) dapat ditulis ulang menjadi satu kesatuan sistem persamaan dan ditulis dalam notasi matriks sebagai berikut,

$$\begin{aligned} \mathbf{t}(k + 1) = & [A_0 \otimes \mathbf{t}(k + 1)] \oplus [A_1 \otimes \mathbf{t}(k)] \oplus [A_2 \otimes \mathbf{t}(k - 1)] \oplus \\ & [A_3 \otimes \mathbf{t}(k - 2)] \oplus [A_4 \otimes \mathbf{t}(k - 3)] \end{aligned} \quad (4.13)$$

dengan,

$$\mathbf{t}(k + 1) = [t_1(k + 1) \quad t_2(k + 1) \quad t_3(k + 1) \quad \cdots \quad t_{12}(k + 1)]'$$

$$\mathbf{t}(k) = [t_1(k) \quad t_2(k) \quad t_3(k) \quad t_4(k) \quad \cdots \quad t_{12}(k)]'$$

$$\mathbf{t}(k - 1) = [t_1(k - 1) \quad t_2(k - 1) \quad t_3(k - 1) \quad \cdots \quad t_{12}(k - 1)]'$$

$$\mathbf{t}(k - 2) = [t_1(k - 2) \quad t_2(k - 2) \quad t_3(k - 2) \quad \cdots \quad t_{12}(k - 2)]'$$

$$\mathbf{t}(k - 3) = [t_1(k - 3) \quad t_2(k - 3) \quad t_3(k - 3) \quad \cdots \quad t_{12}(k - 3)]'$$

dan,

$$A_0 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ a_{2,1} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & b & \epsilon & \epsilon & \epsilon \\ \epsilon & a_{3,2} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & a_{4,3} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & b & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & a_{5,4} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & a_{6,5} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & a_{8,7} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & a_{9,8} & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & b & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & a_{10,9} & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & a_{11,10} & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & a_{12,11} & \epsilon \end{bmatrix}$$

selain itu,

[illegible]

[illegible]

[illegible]

[illegible]

Untuk keperluan perhitungan nilai eigen, selanjutnya diinputkan nilai parameter waktu tempuh yang diperlukan bagi kereta api untuk berjalan dari arah WR menuju ke GDG yaitu $a_{3,2} = 6$, dan dari arah yang berlawanan atau dari GDG menuju ke WR yaitu $a_{11,10} = 6$ menit, GDG menuju ke SDA yaitu $a_{5,4} = 10$, dan dari arah yang berlawanan atau dari SDA menuju ke GDG yaitu $a_{9,8} = 10$ menit, kemudian waktu tunggu kereta api atau berhenti di masing-masing stasiun yaitu $a_{2,1} = a_{4,3} = a_{6,5} = a_{8,7} = a_{10,9} = a_{12,11} = 2$ menit, dan lama waktu tunggu kereta api untuk berjalan setelah kedatangan kereta api dari arah berlawanan yaitu $b = 1$ menit, serta waktu yang diperlukan bagi kereta api yang berjalan dari luar sistem yaitu $a_1 = a_7 = 0$ menit.

Nilai-nilai parameter tersebut didapatkan dari data keberangkatan kereta api jalur Waru-Sidoarjo tertanggal 31 Januari 2015 yang diambil dari PT. Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya (lihat Lampiran). Data tersebut merupakan data keberangkatan kereta api reguler, dan nilai parameter yang diperoleh merupakan hasil perhitungan rata-rata waktu tempuh dari keseluruhan jadwal keberangkatan kereta api dalam durasi satu hari dan hasil perhitungan rata-rata tersebut dibulatkan ke bilangan bulat terdekat, karena dalam hal ini, bilangan di belakang koma merupakan nilai yang menyatakan waktu tempuh yang digunakan kereta api dalam satuan detik, jadi dalam penelitian ini tidak terlalu diperhitungkan.

Setelah memasukkan nilai-nilai parameter yang didapatkan di atas ke dalam Persamaan (4.13), diketahui bahwa A_0 yaitu

$$A_0 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon \\ \epsilon & 6 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 9 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 9 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 6 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2 & \epsilon \end{bmatrix}$$

merupakan matriks yang mempunyai bobot sirkuit rata-rata kurang dari atau sama dengan e , atau tidak mempunyai sirkuit, sehingga bisa didapatkan A_0^* yaitu

$$A_0^* = \begin{bmatrix} 0 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 2 & 0 & \epsilon & \epsilon & \epsilon & \epsilon & 13 & 11 & 1 & \epsilon & \epsilon & \epsilon \\ 8 & 6 & 0 & \epsilon & \epsilon & \epsilon & 19 & 17 & 7 & \epsilon & \epsilon & \epsilon \\ 10 & 8 & 2 & 0 & \epsilon & \epsilon & 21 & 19 & 9 & \epsilon & \epsilon & \epsilon \\ 20 & 18 & 12 & 10 & 0 & \epsilon & 31 & 29 & 19 & \epsilon & \epsilon & \epsilon \\ 22 & 20 & 14 & 12 & 2 & 0 & 33 & 31 & 21 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 0 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2 & 0 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 12 & 10 & 0 & \epsilon & \epsilon & \epsilon \\ 9 & 7 & 1 & \epsilon & \epsilon & \epsilon & 20 & 18 & 8 & 0 & \epsilon & \epsilon \\ 15 & 13 & 7 & \epsilon & \epsilon & \epsilon & 26 & 24 & 14 & 6 & 0 & \epsilon \\ 17 & 15 & 9 & \epsilon & \epsilon & \epsilon & 28 & 26 & 16 & 8 & 2 & 0 \end{bmatrix}$$

Kemudian, berdasarkan **Teorema 2.2**, maka Persamaan (4.13) dapat dikonstruksi ulang seperti pada Persamaan (2.12) menjadi,

$$\begin{aligned} \mathbf{t}(k+1) = & [(A_0^* \otimes A_1) \otimes \mathbf{t}(k)] \oplus [(A_0^* \otimes A_2) \otimes \mathbf{t}(k-1)] \oplus [(A_0^* \otimes A_3) \otimes \\ & \mathbf{t}(k-2)] \oplus [(A_0^* \otimes A_4) \otimes \mathbf{t}(k-3)] \end{aligned} \quad (4.14)$$

dengan,

$$A_0^* \otimes A_1 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon \\ 1 & \epsilon & 12 & \epsilon & 14 & \epsilon & 12 & \epsilon & \epsilon & \epsilon & 3 & \epsilon \\ 7 & \epsilon & 18 & \epsilon & 20 & \epsilon & 18 & \epsilon & \epsilon & \epsilon & 9 & \epsilon \\ 9 & \epsilon & 20 & \epsilon & 22 & \epsilon & 20 & \epsilon & \epsilon & \epsilon & 11 & \epsilon \\ 19 & \epsilon & 30 & \epsilon & 32 & \epsilon & 30 & \epsilon & \epsilon & \epsilon & 21 & \epsilon \\ 21 & \epsilon & 32 & \epsilon & 34 & \epsilon & 32 & \epsilon & \epsilon & \epsilon & 23 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 1 & \epsilon & 3 & \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 11 & \epsilon & 13 & \epsilon & 11 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 8 & \epsilon & 19 & \epsilon & 21 & \epsilon & 19 & \epsilon & \epsilon & \epsilon & 10 & \epsilon \\ 14 & 1 & 25 & \epsilon & 27 & \epsilon & 25 & \epsilon & \epsilon & \epsilon & 16 & \epsilon \\ 16 & 3 & 27 & \epsilon & 29 & \epsilon & 27 & \epsilon & \epsilon & \epsilon & 18 & \epsilon \end{bmatrix}$$

$$A_0^* \otimes A_2 = \begin{bmatrix} \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 3 & \epsilon & 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2 & \epsilon & \epsilon \\ \epsilon & 9 & \epsilon & 8 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 8 & \epsilon & \epsilon \\ \epsilon & 11 & \epsilon & 10 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 10 & \epsilon & \epsilon \\ \epsilon & 21 & \epsilon & 20 & \epsilon & \epsilon & \epsilon & 0 & \epsilon & 20 & \epsilon & \epsilon \\ \epsilon & 23 & \epsilon & 22 & \epsilon & \epsilon & \epsilon & 2 & \epsilon & 22 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon \\ \epsilon & 10 & \epsilon & 9 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 9 & \epsilon & \epsilon \\ \epsilon & 16 & \epsilon & 15 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 15 & \epsilon & 1 \\ \epsilon & 18 & \epsilon & 17 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 17 & \epsilon & 3 \end{bmatrix}$$

selain itu,

$$A_0^* \otimes A_3 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 0 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2 & \epsilon & 14 & \epsilon & \epsilon & \epsilon & 13 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 8 & \epsilon & 20 & \epsilon & \epsilon & \epsilon & 19 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 10 & \epsilon & 22 & \epsilon & \epsilon & \epsilon & 21 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 20 & \epsilon & 32 & \epsilon & \epsilon & \epsilon & 31 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 22 & \epsilon & 34 & \epsilon & \epsilon & \epsilon & 33 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon & 0 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 3 & \epsilon & \epsilon & \epsilon & 2 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 13 & \epsilon & \epsilon & \epsilon & 12 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 9 & \epsilon & 21 & \epsilon & \epsilon & \epsilon & 20 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 15 & \epsilon & 27 & \epsilon & \epsilon & \epsilon & 26 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 17 & \epsilon & 29 & \epsilon & \epsilon & \epsilon & 28 \end{bmatrix}$$

$$A_0^* \otimes A_4 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 14 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 20 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 22 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 32 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 34 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 3 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 13 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 21 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 27 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 29 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

Langkah terakhir, dari Persamaan (4.14) dapat ditransformasikan ke dalam bentuk persamaan relasi rekurensi orde-1 seperti pada Persamaan (2.13) dan didapatkan persamaan relasi rekurensi orde-1 sebagai berikut,

$$\tilde{\mathbf{t}}(k+1) = \tilde{A}(k) \otimes \tilde{\mathbf{t}}(k) \quad (4.15)$$

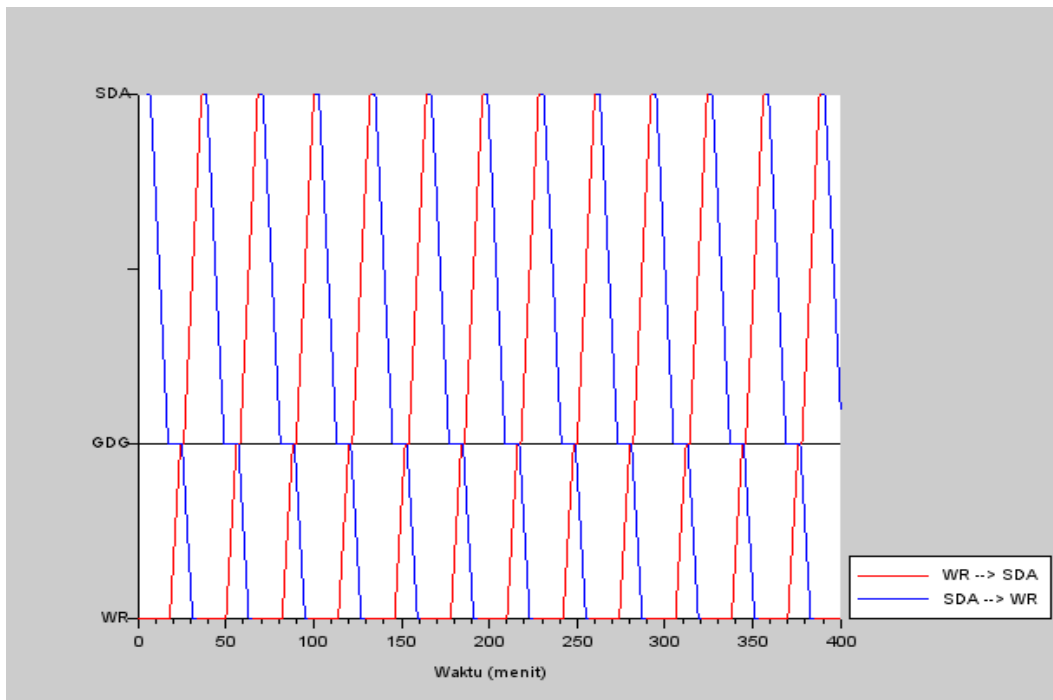
dengan

$$\tilde{A} = \begin{bmatrix} A_0^* \otimes A_1 & A_0^* \otimes A_2 & A_0^* \otimes A_3 & A_0^* \otimes A_4 \\ E & \mathcal{E} & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & E & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & \mathcal{E} & E & \mathcal{E} \end{bmatrix}$$

dan

$$\tilde{\mathbf{t}}(k) = [\mathbf{t}'(k), \mathbf{t}'(k-1), \mathbf{t}'(k-2), \mathbf{t}'(k-3)]'$$

Persamaan (4.15) merupakan model Aljabar Max-Plus yang diinginkan. Untuk selanjutnya mengenai perhitungan nilai eigen dari model Aljabar Max-Plus akan digunakan algoritma power yang terdapat pada *toolbox* Petri Net dan Aljabar Max-Plus pada aplikasi Scilab dan didapatkan bahwa $\lambda = 32$. Angka ini menunjukkan bahwa diperlukan waktu 32 menit bagi kereta api yang berjalan searah untuk berjalan pada siklus berikutnya dari keberangkatan sebelumnya pada masing-masing stasiun. Berikut hasil penjadwalan kedatangan dan keberangkatan kereta api untuk 400 menit pertama atau 6 jam dan 40 menit pertama, dapat dilihat bahwa pada selang waktu tersebut dapat digunakan untuk memberangkatkan 24 kereta api, yaitu 12 kereta api dengan arah SDA-WR dan 12 kereta api dengan arah WR-SDA (lihat Gambar 4.17).



Gambar 4.17 Waktu Kedatangan dan Keberangkatan Kereta Api 400 Menit Pertama

Sedangkan untuk jadwal kedatangan dan keberangkatan kereta api dapat dilakukan konversi waktu dari format menit menjadi format jam. Jika keberangkatan pertama dilakukan pada pukul 03:54, maka didapatkan jadwal kedatangan dan keberangkatan kereta api pada masing-masing stasiun sebagai berikut (lihat Tabel 4.1 dan Tabel 4.2). Tabel 4.1 merupakan jadwal kedatangan

dan keberangkatan kereta api yang berjalan dari arah WR menuju ke SDA, sedangkan Tabel 4.2 merupakan jadwal kedatangan dan keberangkatan kereta api yang berjalan dari arah SDA menuju ke WR. Berdasarkan tabel tersebut dapat dilihat bahwa dalam selang waktu 19 jam dan 50 menit, yaitu mulai pukul 03:54 sampai pukul 23:44 dapat digunakan untuk memberangkatkan 74 kereta api, 37 kereta api berjalan dari arah WR menuju ke SDA dan 37 kereta api berjalan dari arah SDA menuju ke WR. Pertemuan kereta api yang berjalan berlawanan arah selalu berada di stasiun, menandakan bahwa tidak terjadi tabrakan.

Tabel 4.1 Jadwal Kedatangan dan Keberangkatan Kereta Api dari Arah WR Menuju ke SDA

No	Waru		Gedangan		Sidoarjo	
	Kedatangan	Keberangkatan	Kedatangan	Keberangkatan	Kedatangan	Keberangkatan
1	3:54	4:12	4:18	4:20	4:30	4:32
2	4:26	4:44	4:50	4:52	5:02	5:04
3	4:58	5:16	5:22	5:24	5:34	5:36
4	5:30	5:48	5:54	5:56	6:06	6:08
5	6:02	6:20	6:26	6:28	6:38	6:40
6	6:34	6:52	6:58	7:00	7:10	7:12
7	7:06	7:24	7:30	7:32	7:42	7:44
8	7:38	7:56	8:02	8:04	8:14	8:16
9	8:10	8:28	8:34	8:36	8:46	8:48
10	8:42	9:00	9:06	9:08	9:18	9:20
11	9:14	9:32	9:38	9:40	9:50	9:52
12	9:46	10:04	10:10	10:12	10:22	10:24
13	10:18	10:36	10:42	10:44	10:54	10:56
14	10:50	11:08	11:14	11:16	11:26	11:28
15	11:22	11:40	11:46	11:48	11:58	12:00
16	11:54	12:12	12:18	12:20	12:30	12:32
17	12:26	12:44	12:50	12:52	13:02	13:04
18	12:58	13:16	13:22	13:24	13:34	13:36
19	13:30	13:48	13:54	13:56	14:06	14:08
20	14:02	14:20	14:26	14:28	14:38	14:40
21	14:34	14:52	14:58	15:00	15:10	15:12
22	15:06	15:24	15:30	15:32	15:42	15:44
23	15:38	15:56	16:02	16:04	16:14	16:16
24	16:10	16:28	16:34	16:36	16:46	16:48
25	16:42	17:00	17:06	17:08	17:18	17:20

No	Waru		Gedangan		Sidoarjo	
	Kedatangan	Keberangkatan	Kedatangan	Keberangkatan	Kedatangan	Keberangkatan
26	17:14	17:32	17:38	17:40	17:50	17:52
27	17:46	18:04	18:10	18:12	18:22	18:24
28	18:18	18:36	18:42	18:44	18:54	18:56
29	18:50	19:08	19:14	19:16	19:26	19:28
30	19:22	19:40	19:46	19:48	19:58	20:00
31	19:54	20:12	20:18	20:20	20:30	20:32
32	20:26	20:44	20:50	20:52	21:02	21:04
33	20:58	21:16	21:22	21:24	21:34	21:36
34	21:30	21:48	21:54	21:56	22:06	22:08
35	22:02	22:20	22:26	22:28	22:38	22:40
36	22:34	22:52	22:58	23:00	23:10	23:12
37	23:06	23:24	23:30	23:32	23:42	23:44

Pada hasil penjadwalan Tabel 4.1 dan Tabel 4.2 jika dikaitkan dengan jadwal keberangkatan kereta api yang diambil dari PT. Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya tertanggal 31 Januari 2015 (lihat Lampiran), maka dapat disimpulkan bahwa keberangkatan kereta api barang yang berjalan dari arah SDA menuju ke WR merupakan keberangkatan yang ke-2 dengan selisih waktu maksimal 20 menit dari jadwal keberangkatan sebenarnya. Dapat dilihat pada Tabel 4.2 keberangkatan ke-2, pada tabel tersebut menunjukkan bahwa kedatangan kereta api di SDA yaitu pada pukul 04:31, sedangkan pada jadwal sebenarnya pukul 04:20 atau selisih 11 menit, dan keberangkatan kereta api di SDA yaitu pada pukul 04:33, sedangkan pada jadwal sebenarnya pukul 04:20 atau selisih 13 menit. Kemudian kedatangan kereta api di GDG yaitu pada pukul 04:43, sedangkan pada jadwal sebenarnya pukul 04:31 atau selisih 12 menit, dan keberangkatan kereta api di GDG yaitu pada pukul 04:51, sedangkan pada jadwal sebenarnya pukul 04:31 atau selisih 20 menit. Selain itu kedatangan kereta api di WR yaitu pada pukul 04:57, sedangkan pada jadwal sebenarnya pukul 04:38 atau selisih 19 menit, dan keberangkatan kereta api di WR yaitu pada pukul 04:59, sedangkan pada jadwal sebenarnya pukul 04:59 atau tidak ada selisih.

Tabel 4.2 Jadwal Kedatangan dan Keberangkatan Kereta Api dari Arah SDA Menuju ke WR

No	Sidoarjo		Gedangan		Waru	
	Kedatangan	Keberangkatan	Kedatangan	Keberangkatan	Kedatangan	Keberangkatan
1	3:59	4:01	4:11	4:19	4:25	4:27
2	4:31	4:33	4:43	4:51	4:57	4:59
3	5:03	5:05	5:15	5:23	5:29	5:31
4	5:35	5:37	5:47	5:55	6:01	6:03
5	6:07	6:09	6:19	6:27	6:33	6:35
6	6:39	6:41	6:51	6:59	7:05	7:07
7	7:11	7:13	7:23	7:31	7:37	7:39
8	7:43	7:45	7:55	8:03	8:09	8:11
9	8:15	8:17	8:27	8:35	8:41	8:43
10	8:47	8:49	8:59	9:07	9:13	9:15
11	9:19	9:21	9:31	9:39	9:45	9:47
12	9:51	9:53	10:03	10:11	10:17	10:19
13	10:23	10:25	10:35	10:43	10:49	10:51
14	10:55	10:57	11:07	11:15	11:21	11:23
15	11:27	11:29	11:39	11:47	11:53	11:55
16	11:59	12:01	12:11	12:19	12:25	12:27
17	12:31	12:33	12:43	12:51	12:57	12:59
18	13:03	13:05	13:15	13:23	13:29	13:31
19	13:35	13:37	13:47	13:55	14:01	14:03
20	14:07	14:09	14:19	14:27	14:33	14:35
21	14:39	14:41	14:51	14:59	15:05	15:07
22	15:11	15:13	15:23	15:31	15:37	15:39
23	15:43	15:45	15:55	16:03	16:09	16:11
24	16:15	16:17	16:27	16:35	16:41	16:43
25	16:47	16:49	16:59	17:07	17:13	17:15
26	17:19	17:21	17:31	17:39	17:45	17:47
27	17:51	17:53	18:03	18:11	18:17	18:19
28	18:23	18:25	18:35	18:43	18:49	18:51
29	18:55	18:57	19:07	19:15	19:21	19:23
30	19:27	19:29	19:39	19:47	19:53	19:55
31	19:59	20:01	20:11	20:19	20:25	20:27
32	20:31	20:33	20:43	20:51	20:57	20:59
33	21:03	21:05	21:15	21:23	21:29	21:31
34	21:35	21:37	21:47	21:55	22:01	22:03
35	22:07	22:09	22:19	22:27	22:33	22:35
36	22:39	22:41	22:51	22:59	23:05	23:07
37	23:11	23:13	23:23	23:31	23:37	23:39

Begitu juga pada keberangkatan kereta api barang yang berjalan dari arah WR menuju ke SDA. Berdasarkan hasil penjadwalan yang didapatkan pada Tabel 4.1 jika dibandingkan dengan jadwal sebenarnya dapat disimpulkan bahwa keberangkatan kereta api barang yang berjalan dari arah WR menuju ke SDA merupakan keberangkatan yang ke-18 dengan selisih waktu maksimal 10 menit dari jadwal keberangkatan sebenarnya. Dapat dilihat pada Tabel 4.1 keberangkatan ke-18, pada tabel tersebut menunjukkan bahwa kedatangan kereta api di WR yaitu pada pukul 12:58, sedangkan pada jadwal sebenarnya pukul 13:08 atau selisih 10 menit, sedangkan keberangkatan kereta api di WR yaitu pada pukul 13:16, sedangkan pada jadwal sebenarnya pukul 13:08 atau selisih 8 menit. Kemudian kedatangan kereta api di GDG yaitu pada pukul 13:22, sedangkan pada jadwal sebenarnya pukul 13:15 atau selisih 7 menit, sedangkan keberangkatan kereta api di GDG yaitu pada pukul 13:24, sedangkan pada jadwal sebenarnya pukul 13:20 atau selisih 4 menit. Selain itu kedatangan kereta api di SDA yaitu pada pukul 13:34, sedangkan pada jadwal sebenarnya pukul 13:35 atau selisih 1 menit, sedangkan keberangkatan kereta api di SDA yaitu pada pukul 13:36, sedangkan pada jadwal sebenarnya pukul 13:35 atau selisih 1 menit.

Secara umum akan terjadi selisih waktu antara jadwal hasil perhitungan dengan jadwal yang diambil dari PT. Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya. Selisih waktu tersebut dikarenakan pemberian waktu tunggu di masing-masing stasiun yang disamaratakan yaitu berdasarkan waktu tunggu rata-rata, begitu juga pada waktu tempuh yang diperlukan kereta api untuk berjalan dari satu stasiun ke stasiun berikutnya. Selain itu dikarenakan waktu tunggu yang diberikan pada keberangkatan setiap kereta api harus menunggu selama 1 menit setelah kedatangan kereta api yang berjalan dari arah yang berlawanan.

Pada jadwal yang diperoleh dari PT. Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya, terdapat jam-jam kosong seperti pada pukul 06:30 sampai dengan pukul 08:15 dan pada pukul 09:25 sampai dengan pukul 11:35, sedangkan pada perhitungan diperoleh hasil penjadwalan yang kontinu untuk setiap waktunya. Tambahan jadwal keberangkatan pada hasil perhitungan ini bisa direkomendasikan untuk digunakan keberangkatan kereta api yang tidak masuk

dalam jadwal sehari-hari, sebagai contoh bisa digunakan untuk keberangkatan kereta api Sewa/Carter (Paket Rombongan) dan kereta api Wisata. Selain itu bisa digunakan untuk penambahan keberangkatan kereta api, seperti pada penambahan keberangkatan kereta api karena adanya kelonjakan penumpang pada musim lebaran (arus mudik), atau penambahan keberangkatan kereta api seperti yang dilakukan pada tanggal 1 April 2015, mulai beroperasi KA Ekonomi Sidoarjo-Surabaya Gubeng-Surabaya Pasarturi-Bojonegoro (PP) yang sebelumnya merupakan layanan rute Surabaya Pasarturi-Bojonegoro (PP). Pada dasarnya sudah terdapat jalur yang menghubungkan Surabaya Gubeng dengan Surabaya Pasarturi sebelumnya, hanya saja tidak dibuat jadwal keberangkatan antara stasiun tersebut. Penambahan keberangkatan kereta api ini dilakukan untuk memudahkan pengguna jasa kereta api dari Sidoarjo yang ingin menuju ke Surabaya Pasarturi, atau dari Bojonegoro yang ingin menuju ke Sidoarjo.

Selain itu pada jadwal keberangkatan yang diperoleh dari hasil perhitungan nilai eigen dari model Aljabar Max-Plus didapatkan sifat keperiodikan yang seragam, yaitu keberangkatan kereta api terjadi setiap 32 menit sekali pada masing-masing stasiun untuk kereta api yang berjalan dengan arah yang sama. Hal ini merupakan keperiodikan minimum yang diperoleh dari model Aljabar Max-Plus yang dibangun untuk mendapatkan jadwal keberangkatan yang efisien dari segi waktu dan penggunaan jalur kereta api *semi-double track* sehingga tidak terjadi *bottleneck*.

BAB 5

KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah diberikan, dapat dibuat kesimpulan serta saran untuk pengembangan dan perbaikan penelitian selanjutnya.

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian dan pembahasan yang telah diberikan adalah sebagai berikut:

1. Struktur hirarkis jalur kereta api *semi-double track* merupakan aturan yang diberikan pada sistem sehingga tidak terjadi *bottleneck*.
2. Model Petri Net yang didapatkan merupakan implementasi dari struktur hirarkis jalur kereta api *semi-double track*, sehingga dalam model Petri Net tersebut tidak terjadi *deadlock*.
3. Model Aljabar Max-Plus yang didapatkan dari model Petri Net merupakan persamaan rekurensi orde- n dengan $n > 1$, sehingga perlu ditransformasikan ke dalam bentuk persamaan rekurensi orde-1 untuk mendapatkan nilai eigen (sifat keperiodikan) dari model Aljabar Max-Plus yang didapatkan.
4. Berdasarkan hasil perhitungan nilai eigen dari model Aljabar Max-Plus didapatkan bahwa $\lambda = 32$, artinya diperlukan waktu 32 menit bagi kereta api untuk berjalan pada siklus berikutnya dari keberangkatan sebelumnya yang berjalan searah pada masing-masing stasiun. Selain itu, dapat disimpulkan bahwa jika keberangkatan awal kereta api dimulai pukul 03:54 dan diakhiri sampai pukul 23:44, maka ada 74 kereta api yang dapat diberangkatkan, yaitu 37 kereta api berjalan dari arah WR menuju ke SDA dan 37 kereta api berjalan dari arah SDA menuju ke WR. Secara umum akan terjadi selisih waktu antara jadwal hasil perhitungan dengan jadwal yang diambil dari PT. Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya. Selisih waktu tersebut dikarenakan pemberian waktu tunggu di masing-masing stasiun yang disamaratakan yaitu berdasarkan waktu tunggu rata-rata, begitu juga pada

waktu tempuh yang diperlukan kereta api untuk berjalan dari satu stasiun ke stasiun berikutnya. Diketahui bahwa pertemuan kereta api yang berjalan berlawanan arah selalu berada di stasiun, menandakan bahwa tidak terjadi tabrakan antara kereta api yang berjalan dari arah SDA menuju ke WR dengan kereta api yang berjalan dari arah WR menuju ke SDA.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, saran yang dapat diberikan adalah bagaimana sifat keperiodikan dari sistem yang dapat dibuat jika diterapkan pada ruang lingkup penelitian yang lebih luas lagi (jalur kereta api yang lebih kompleks, yaitu dengan adanya jalur *double track* yang menghubungkan dua stasiun berdampingan).

DAFTAR PUSTAKA

- Abdulkadir, M., (1998), *Hukum Pengangkutan Niaga*, Bandung, Citra Aditya.
- Afiatna, F. A., (2013), *Pemodelan dan Penjadwalan Monorel dan Trem yang Terintegrasi di Kota Surabaya Menggunakan Aljabar Max-Plus*, Tugas Akhir S1 Matematika, Institut Teknologi Sepuluh Nopember Surabaya.
- Alfiah, S., (2011), *Pemodelan Jaringan Kereta Rel Listrik (KRL) Menggunakan Aljabar Max-Plus*, Tesis Magister Matematika, Institut Teknologi Sepuluh Nopember Surabaya.
- Anggraeni, D. A., (2013), *Analisis Jadwal Keberangkatan Pesawat Transit di Bandara dengan Menggunakan Aljabar Max-Plus*, Tugas Akhir S1 Matematika, Institut Teknologi Sepuluh Nopember Surabaya.
- Badan Pusat Statistik, (2014), *Statistik Indonesia 2014*, Diambil kembali dari <http://www.bps.go.id/int/index.php/site/search?search=pdb>.
- Cassandras, C. G., & Lafortune, S., (2008), *Introduction to Discrete Event Systems Second Edition*, New York, Springer.
- Direktorat Jenderal Perhubungan Darat, (2014), *Perhubungan Darat dalam Angka 2013*, Jakarta, Kementrian Perhubungan RI.
- Fahim, K., (2013), *Aplikasi Aljabar Max-Plus pada Pemodelan dan Penjadwalan Busway yang Diintegrasikan dengan Kereta Api Komuter*, Tugas Akhir S1 Matematika, Institut Teknologi Sepuluh Nopember Surabaya.
- Heidergott, B., Olsder, G. J., & Woude, J. V., (2006), *Max Plus at Work, Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*, United Kingdom, Princeton University Press.
- Li, Danjing, (2008), *A Hierarchical Control Structure for A Class of Timed Discrete Event Systems*, Dissertation of Electrical Engineering and Computer Science, Technical University of Berlin, Magdeburg.
- Oktafianto, K., (2013), *Implementasi Aljabar Max-Plus pada Pemodelan dan Penjadwalan Keberangkatan Bus Kota Damri (Studi Kasus di Surabaya)*, Tugas Akhir S1 Matematika, Institut Teknologi Sepuluh Nopember Surabaya.

PT Kereta Api Indonesia (Persero), (2014, Oktober 19), Diambil kembali dari PT Kereta Api Indonesia (Persero), <http://kereta-api.co.id>.

Subiono, (2015), *Aljabar Min-Max Plus dan Terapannya Version 3.0.0.*, Buku Ajar Mata Kuliah Pilihan Pascasarjana Matematika Institut Teknologi Sepuluh Nopember Surabaya.

LAMPIRAN

A. Jadwal Keberangkatan Kereta Api Tanggal 31 Januari 2015 Waru-Sidoarjo (PT. Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya)

1. Pagi

No	No. KA	Waru		Gedangan		Sidoarjo	
		Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an
1	305	4:09	4:15	4:21	4:33	4:42	4:44
2	207	4:41	4:43	4:48	4:48	4:56	4:58
3	439	4:56	4:58	5:04	5:05	5:15	5:19
4	43	6:11	6:11	6:15	6:15	6:23	6:26

2. Siang

No	No. KA	Waru		Gedangan		Sidoarjo	
		Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an
1	441	8:21	8:32	8:38	8:40	8:49	8:53
2	85	9:11	9:11	9:15	9:15	9:23	9:25
3	443	11:37	11:39	11:45	11:47	11:56	12:02
4	(*)2613	13:08	13:08	13:15	13:20	13:35	13:35
5	175	13:50	13:50	13:54	13:54	14:02	14:04
6	313	14:09	14:11	14:19	14:32	14:47	14:54
7	169	14:56	14:56	15:01	15:07	15:16	15:19

(*) Kereta Barang

3. Malam

No	No. KA	Waru		Gedangan		Sidoarjo	
		Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an
1	447	16:16	16:18	16:24	16:25	16:34	16:36
2	2615	17:07	17:14	17:23	17:23	17:26	17:26
3	317	18:08	18:11	18:20	18:25	18:40	18:42
4	449	20:36	20:38	20:43	20:43	20:51	20:53
5	87	22:23	22:23	22:27	22:27	22:35	22:37
6	KP/7073a	23:34	23:34	23:38	23:38	23:47	23:50

B. Jadwal Keberangkatan Kereta Api Tanggal 31 Januari 2015 Sidoarjo-Waru (PT. Kereta Api Indonesia (Persero) Daerah Operasi 8 Surabaya)

1. Pagi

No	No. KA	Sidoarjo		Gedangan		Waru	
		Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an
1	88	3:54	3:59	4:08	4:08	4:13	4:13
2	(*)2614	4:20	4:20	4:31	4:31	4:38	4:59
3	438	5:17	5:20	5:29	5:31	5:37	5:39
4	306	5:56	5:58	6:11	6:17	6:25	6:26

(*) Kereta Barang

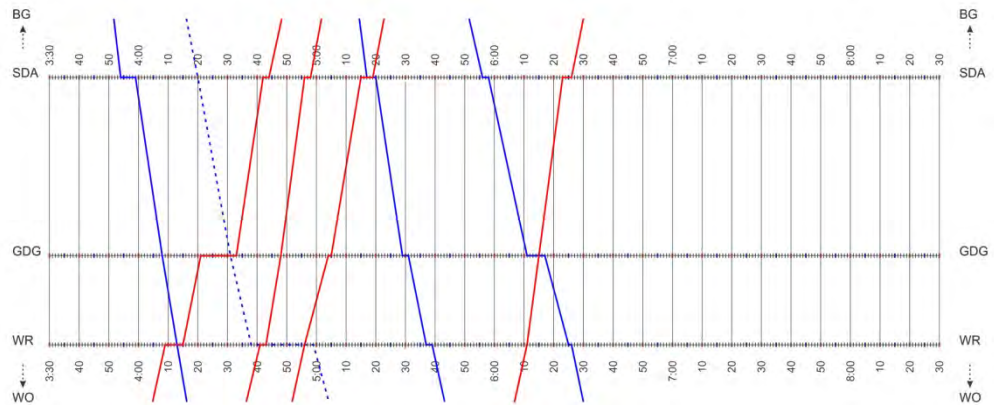
2. Siang

No	No. KA	Sidoarjo		Gedangan		Waru	
		Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an
1	170	8:15	8:18	8:26	8:26	8:30	8:30
2	440	8:52	8:56	9:04	9:17	9:23	9:25
3	KP/7074a	12:22	12:24	12:34	12:34	12:38	12:38
4	176	12:35	12:37	12:45	12:45	12:49	12:49
5	2616	13:10	13:10	13:19	13:19	13:24	13:24
6	444	14:19	14:21	14:31	14:31	14:37	14:38
7	86	14:52	14:56	15:05	15:05	15:09	15:09
8	44	15:11	15:16	15:26	15:26	15:30	15:30

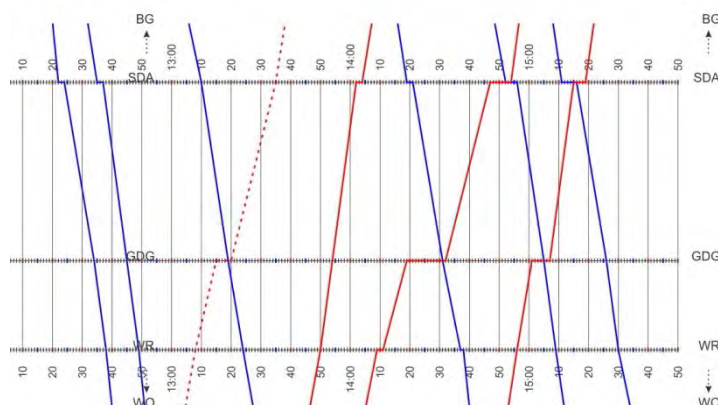
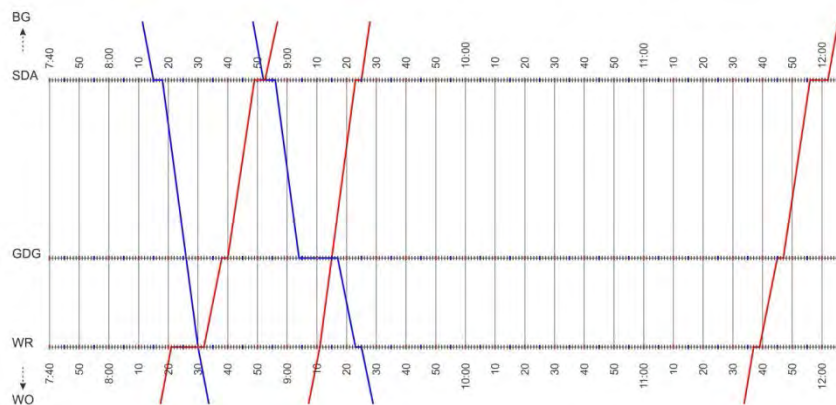
3. Malam

No	No. KA	Sidoarjo		Gedangan		Waru	
		Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an	Ke-datang-an	Ke-berangkat-an
1	314	16:03	16:05	16:20	16:26	16:34	16:39
2	446	19:02	19:04	19:13	19:19	19:25	19:28
3	318	19:42	19:44	19:59	20:00	20:08	20:08
4	208	21:20	21:22	21:30	21:30	21:35	21:37
5	448	21:56	21:58	22:06	22:06	22:11	22:25

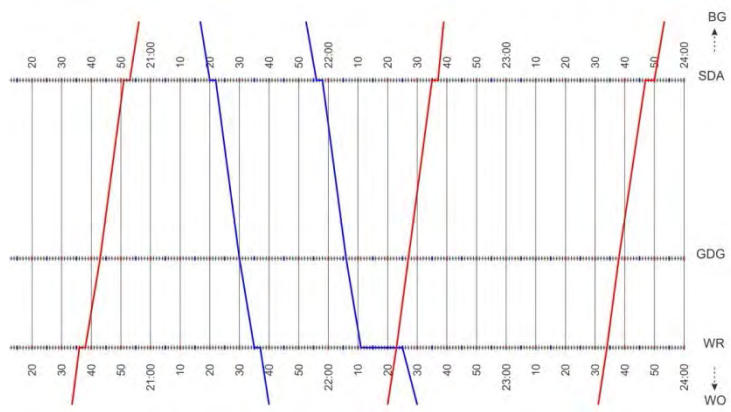
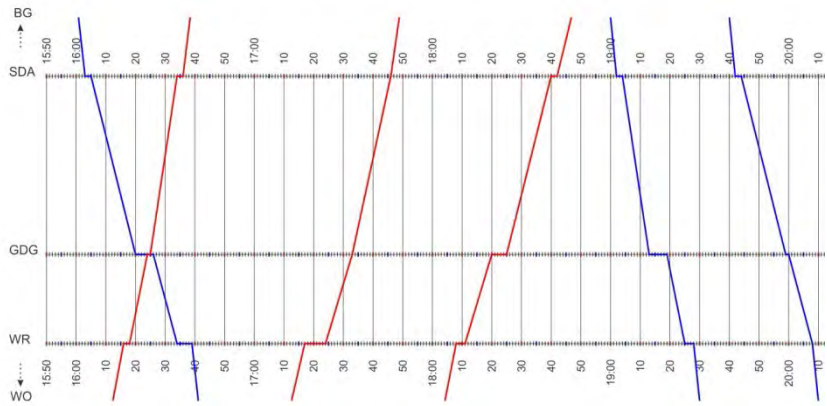
1. Pagi



BG



3. Malam



D. Program Perhitungan Model Aljabar Max-Plus Menggunakan Scilab

5.5.0

No	Program
1	<code>clc;</code>
2	
3	<code>n = 12</code>
4	
5	<code>wt_kereta = 1</code>
6	<code>wt_stasiun = 2</code>
7	<code>wt_wr = 0</code>
8	<code>wt_sda = 0</code>
9	
10	<code>wt_wr_gdg = 6</code>
11	<code>wt_gdg_sda = 10</code>
12	<code>wt_sda_gdg = 10</code>
13	<code>wt_gdg_wr = 6</code>
14	
15	<code>vt(1) = wt_wr</code>
16	<code>vt(2) = wt_stasiun</code>
17	<code>vt(3) = wt_wr_gdg</code>
18	<code>vt(4) = wt_stasiun</code>
19	<code>vt(5) = wt_gdg_sda</code>
20	<code>vt(6) = wt_stasiun</code>
21	
22	<code>vt(7) = wt_sda</code>
23	<code>vt(8) = wt_stasiun</code>
24	<code>vt(9) = wt_sda_gdg</code>
25	<code>vt(10) = wt_stasiun</code>
26	<code>vt(11) = wt_gdg_wr</code>
27	<code>vt(12) = wt_stasiun</code>
28	
29	<code>A0 = maxpluszeros(n, n)</code>
30	<code>for i = 1 : 11</code>
31	<code> if i ~= 6 then</code>
32	<code> A0(i+1, i) = vt(i+1)</code>
33	<code> end</code>
34	<code>end</code>
35	<code>A0(2, [9]) = wt_kereta</code>
36	<code>A0(4, [9]) = wt_kereta</code>
37	<code>A0(10, [3]) = wt_kereta</code>
38	
39	<code>A1 = maxpluszeros(n, n)</code>
40	<code>A1(1, [11]) = wt_kereta</code>
41	<code>A1(2, [11]) = wt_kereta</code>
42	<code>A1(7, [5]) = wt_kereta</code>
43	<code>A1(8, [3,5]) = wt_kereta</code>
44	<code>A1(11, [2]) = wt_kereta</code>
45	
46	<code>A2 = maxpluszeros(n, n)</code>
47	<code>A2(1, [2]) = wt_kereta</code>
48	<code>A2(3, [4,10]) = wt_kereta</code>
49	<code>A2(5, [8]) = wt_kereta</code>
50	<code>A2(9, [4,10]) = wt_kereta</code>
51	<code>A2(11, [12]) = wt_kereta</code>
52	

```

53 A3 = maxpluszeros(n, n)
54 A3(1, [12]) = wt_kereta
55 A3(5, [6]) = wt_kereta
56 A3(7, [8]) = wt_kereta
57
58 A4 = maxpluszeros(n, n)
59 A4(7, [6]) = wt_kereta
60
61 A3(1, [6]) = vt(1)
62 A3(7, [12]) = vt(7)
63
64 A1(2, [1]) = wt_kereta
65 A1(8, [7]) = wt_kereta
66
67 A0_s = maxplusstar(A0)
68
69 A1_1 = (maxplusotimes(A0_s,A1))
70 A2_1 = (maxplusotimes(A0_s,A2))
71 A3_1 = (maxplusotimes(A0_s,A3))
72 A4_1 = (maxplusotimes(A0_s,A4))
73
74 A_1 = maxpluszeros(n,n);
75 A_2 = maxpluseye(n,n);
76
77 A=[A1_1 A2_1 A3_1 A4_1; A_2 A_1 A_1 A_1; A_1 A_2 A_1 A_1; A_1 A_1 A_2
78 A_1]
79
80 [l,v,d] = maxplusmaxalgol(A)
81
82 disp(A0, 'A0 = ')
83 disp(A1, 'A1 = ')
84 disp(A2, 'A2 = ')
85 disp(A3, 'A3 = ')
86 disp(A4, 'A4 = ')
87
88 disp(A0_s, 'A0_s = ')
89 disp(A1_1, 'A1_1 = ')
90 disp(A2_1, 'A2_1 = ')
91 disp(A3_1, 'A3_1 = ')
92 disp(A4_1, 'A4_1 = ')
93
94 disp(l, 'Nilai Eigen = ')
95
96 h = [v(1:12) v(13:24) v(25:36) v(37:48)]
97 disp(h, 'T(k) T(k-1) T(k-2) T(k-3)')
98
99 n_v = 40
100 hasil = zeros(size(v,1), n_v)
101 hasil(:,1) = v;
102 for i = 2 : n_v
103     hasil(:,i) = maxplusotimes(A, hasil(:,i-1));
104 end
105
106 baris = 1
107 Wr_Sda1 = zeros(1,6)
108 Sda_Wr1 = zeros(1,6)
109 for j = 1 : size(hasil,2)

```

```

110   Wr_Sda1(baris, 1 : 6) = hasil( 1 : 6, j)'
111   Sda_Wr1(baris, 1 : 6) = hasil( 7 : 12, j)'
112   baris = baris + 1
113 end
114
115 pengurang1 = min(min(Wr_Sda1,Sda_Wr1))
116
117 x_Wr_Sda1 = Wr_Sda1-pengurang1
118 x_Sda_Wr1 = Sda_Wr1-pengurang1
119
120 jam_awal_operasi = 3
121 menit_awal_operasi = 54
122
123 jadwal_utr = struct()
124 jadwal_utr.jam    = jam_awal_operasi + floor((x_Wr_Sda1 + menit_awal_operasi)/60)
125 jadwal_utr.menit  = modulo((x_Wr_Sda1 + menit_awal_operasi), 60)
126
127 jadwal_slt = struct()
128 jadwal_slt.jam    = jam_awal_operasi + floor((x_Sda_Wr1 + menit_awal_operasi)/60)
129 jadwal_slt.menit  = modulo((x_Sda_Wr1 + menit_awal_operasi), 60)
130
131 dok_utr = "
132 dok_slt = "
133 jadwal_jadi = cell(1,3)
134 for i= 1 : size(jadwal_utr.jam,1)
135     hsl = "
136     for j = 1 : size(jadwal_utr.jam,2)
137         if jadwal_utr.jam(i,j)<10 then
138             str1 = '0' + string(jadwal_utr.jam(i,j))
139         else
140             str1 = string(jadwal_utr.jam(i,j))
141         end
142         if jadwal_utr.menit(i,j)<10 then
143             str2 = '0' + string(jadwal_utr.menit(i,j))
144         else
145             str2 = string(jadwal_utr.menit(i,j))
146         end
147         dok_utr(i,j)=[str1 + ':' + str2]
148         hsl = [hsl + str1 + ':' + str2 + ', ' ]
149     end
150     jadwal_jadi(i,1).entries = [string(i)]
151     jadwal_jadi(i,2).entries = [hsl]
152     if jadwal_utr.jam(i,6)>= 24 then
153         break
154     end
155 end
156
157 for i= 1 : size(jadwal_slt.jam,1)
158     hsl = "
159     for j = 1 : size(jadwal_slt.jam,2)
160         if jadwal_slt.jam(i,j)<10 then
161             str1 = '0' + string(jadwal_slt.jam(i,j))
162         else
163             str1 = string(jadwal_slt.jam(i,j))
164         end
165         if jadwal_slt.menit(i,j)<10 then
166             str2 = '0' + string(jadwal_slt.menit(i,j))

```

```

167     else
168         str2 = string(jadwal_slt.menit(i,j))
169     end
170     dok_slt(i,j)=[str1 + ':' + str2]
171     hsl = [hsl + str1 + ':' + str2 + ', ' ]
172 end
173 jadwal_jadi(i,3).entries = [hsl]
174 if jadwal_slt.jam(i,6)>= 24 then
175     break
176 end
177 end
178
179 y_Wr_Sda1 = [0 0 2 2 6 6]
180 y_Sda_Wr1 = [6 6 2 2 0 0]
181
182 x_max = max(max(x_Wr_Sda1, x_Sda_Wr1))
183 f0 = figure()
184 clf()
185 plot(x_Wr_Sda1(1,:), y_Wr_Sda1(1,:), 'r', x_Sda_Wr1(1,:), y_Sda_Wr1(1,:), 'b')
186 plot((1:x_max), ones(1,x_max)*2, 'k')
187 plot(x_Wr_Sda1, y_Wr_Sda1, 'r', x_Sda_Wr1, y_Sda_Wr1, 'b')
188
189 a=gca();
190 a.data_bounds=[0 0;400 6];
191 a.y_ticks = tlist(["ticks", "locations", "labels"], (0:2:6)', ["WR";"GDG";"";"SDA"]);
192 a.x_label.text="Waktu (menit)"
193 a.box="off";
194
195 hl=legend(['WR --> SDA';'SDA --> WR']);
196 hl.legend_location='out_lower_right'
197
198 data1=[['No', 'WR', 'GDG', 'SDA'];[string([1:size(dok_utr,1)]), ..
199 (string(dok_utr(:,1)) + '--' + string(dok_utr(:,2))), ..
200 (string(dok_utr(:,3)) + '--' + string(dok_utr(:,4))), ..
201 (string(dok_utr(:,5)) + '--' + string(dok_utr(:,6))))]]
202
203 f1 = figure();
204 f1.position=[10,10,420,400]
205 ut = uicontrol("style","table",..
206 "string",data1,..
207 "position",[10 10 400 380],..
208 "tooltipstring","Data from majors towns")
209
210 data2=[['No', 'SDA', 'GDG', 'WR'];[string([1:size(dok_slt,1)]), ..
211 (string(dok_slt(:,1)) + '--' + string(dok_slt(:,2))), ..
212 (string(dok_slt(:,3)) + '--' + string(dok_slt(:,4))), ..
213 (string(dok_slt(:,5)) + '--' + string(dok_slt(:,6))))]]
214
215 f2 = figure();
216 f2.position=[10,10,420,400]
217 ut = uicontrol("style","table",..
218 "string",data2,..
219 "position",[10 10 400 380],..
220 "tooltipstring","Data from majors towns")

```

BIOGRAFI PENULIS



Penulis bernama Tri Utomo, lahir di Ogan Komeriing Ulu Timur, 02 Nopember 1987, merupakan anak ketiga dari 4 bersaudara. Penulis menempuh pendidikan formal di SDN 2 Sumber Agung (OKU Timur), kemudian melanjutkan ke SLTP N 1 Kurungan Nyawa (OKU Timur), dan MA PLUS Raden Paku (Trenggalek). Setelah lulus dari jenjang Sekolah Menengah Atas penulis melanjutkan studi di Jurusan Informatika dan Teknik Komputer Wearnes Education Center (Program Profesi 1 Tahun) Malang tahun 2006. Kemudian melanjutkan studi S1 di Jurusan Matematika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang tahun 2007. Pada masa perkuliahan Penulis aktif di Laboratorium Komputer Matematika dengan mengadakan pelatihan-pelatihan. Selain itu Penulis menjadi Asisten Dosen untuk beberapa mata kuliah, diantaranya Mata Kuliah Kalkulus I, Pengantar Ilmu Komputer, Pemrograman Komputer I dan II, Pemodelan Matematika, dan Analisis Numerik terhitung mulai Tahun Akademik 2009/2010 sampai 2011/2012. Penulis dapat dihubungi melalui email Three1st@Gmail.Com.