Modeling of Queue System Student Tuition Fee Payment using Petri Net and Max-Plus Algebra

Nurwan Nurwan*, Member IAENG, Stenly Pranata, and Isran K. Hasan

***Abstract*—The phenomenon of students who queue to pay single tuition fee is a common occurrence in banks. As the payment process becomes increasingly complex, an effective approach is needed to model and optimize the queue system. To address this issue, Petri net model is used to represent the dynamics of queue system, including interactions between students, payment officers, and the payment verification process. Therefore, study further explored using max-plus algebra and Petri net approaches to model the payment queue system for students' single tuition fee payments at banks. The verification of Petri net model was conducted using coverability tree, ensuring that the resulting model avoided deadlock and met L1-Live condition. Max-plus algebra model was also used to analyze wait times and facilitate decision-making. Subsequently, model of student payment queue system for single tuition fee was expressed in Equation (8). According to this model, the service time for paying the fee from arrival to departure was 3634 seconds, or 1 hour and 34 seconds.**

***Index Terms*—queue system, petri net, max-plus algebra, student tuition fee payment.**

I. Introduction

THE single tuition fee payment queue system is a common activity found in universities. This process includes various stages, from re-registration to payment through bank services, which result in queues. In this context, the use of Petri net models and max-plus algebra can provide a better understanding of the dynamics of the queue system. According to [16], Petri net model is a mathematical tool capable of modeling and analyzing activities within complex systems. This approach is very suited for describing the logical interactions of various activities within a system, offering a clear and structured process flow analysis. Previous studies have used Petri net to model queue scenarios, including [9], [10], and [17].

Aside from Petri net model, max-plus algebra was introduced by [5], to provide a robust mathematical framework for analyzing time-related systems and event sequences. Max-plus algebra is particularly effective in modeling system dynamics through synchronization, allowing thorough system analysis and simulation. The application of this algebra was described in [7], [8], [15] and [13]. Similarly, the use of max-plus algebra for modeling queue system has been discussed in [9] and [10], typically following the linear algebra equation:

$$d(k) = T(k) \otimes d(k-1) \tag{1}$$

where $d(k)$ represents the arrival time vector of the queue, $T(k)$ is a matrix derived from customer service time, and $\otimes$ signifies the max-plus algebra operator.

This study aimed to integrate these two mathematical models to comprehensively understand the single tuition payment queue system in banks. In order to achieve the aim, Petri net model was built for payment queue and uses a coverability tree to analyze its liveness and deadlock state. Furthermore, max-plus algebra model was constructed to analyze the queue's dynamics and determine the total wait times. Numerical simulations are then carried out to determine the time required for tuition payments. The results of this study are expected to enhance the understanding of Petri net and max-plus algebra applications in queue system analysis as well as to provide recommendations for improving service efficiency and student satisfaction.

II. Preliminaries

A. Petri Net

This section describes the fundamental concepts and matrix representations of Petri net, offering straightforward examples. An exploration of the fundamental Petri net concepts has been described in [3].

***Definition** 1:* Petri net consists of 4-tuple (P, T, A, w), where: P represents finite set of places, $P = \{P_1, P_2, \cdots P_n\}$. T indicates finite set of transition $T = \{T_1, T_2, \cdots T_n\}$. A is arc set, $A \subseteq (P \times T) \cup (T \times P)$, and w is the weights function $w : A \rightarrow 1, 2, 3, \cdots$

Petri net is a directed graph (digraph) comprising two node types, namely circles signifying places and rectangles representing transitions. In the context of this study, the places and transitions are linked by arrows. The weight of the arrow from place P_i to transition T_i was expressed as $w(P_i, T_i) = k$, indicating the presence of k arrows from place P_i to transition T_i . A token is positioned in place P_i to state whether or not a condition is met. A transition is fired when an event stated by the transition happens. Typically, this transition firing process happens when all the token in input place is reduced by the arc weight that connects the places. In Petri net, token is depicted by a black dot in a place [17].

***Definition** 2:* Marked Petri net are 5-tuple $(P; T; A; w; x_0)$ where $(P; T; A; w)$ represents the definition of Petri net, and x_0 is the initial marking.

***Definition** 3:* The state of the marked Petri net is $x = [x(P_1); x(P_2); \cdots ; x(P_n)]^T$

***Definition** 4:* Transition $T_j \in T$ in marked Petri net is said to be enable if $x(P_i) \geq w(P_i, T_j); \qquad \forall P_i \in I(T_j)$

Manuscript received May 27, 2024; revised February 8, 2025

Nurwan Nurwan is a lecturer at the Department of Mathematics, Faculty of Mathematics and Natural Science, Universitas Negeri Gorontalo, Bone Bolango, 96119, Indonesia. (corresponding author to provide e-mail: nurwan@ung.ac.id)

Stenly Pranata is an undergraduate student at the Department of Mathematics, Universitas Negeri Gorontalo, Bone Bolango, 96119, Indonesia. (e-mail: stenlypranata@gmail.com)

Isran K. Hasan is a lecturer in the Department of Statistics, Universitas Negeri Gorontalo, Bone Bolango, 96119, Indonesia. (e-mail: isran.hasan@ung.ac.id)

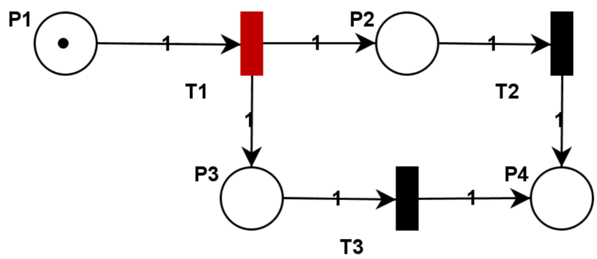

Fig. 1. Simple Petri net

Example 1: A simple Petri net consists of 4 places (P_1, P_2, P_3, P_4) and 3 transitions (T_1, T_2, T_3). The initial state of Petri net in Fig. 1 was $x_0 = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$. The enabled transition is T_1 because it satisfied the following Definition 4.

B. Matrix Representation from Petri Net

This section focused on the matrix representation of Petri net, and the two primary types of matrix derived from Petri net design were backward and forward [3] and [1].

Definition 5: Backward matrix incidence and forward matrix incidence that represents Petri net are $n \times m$ matrix, where n and m is the number of places and transitions consecutively, while the i*th* and j*th* element are $M_b(i,j) = w(p_i, t_j)$ and $M_f(i,j) = w(t_j, p_i)$ respectively.

Typically, backward matrix incidence (A_b) assisted in identifying enabled transitions. In a situation where P_i did not serve as an input place for transition T_j, the weight from P_i to T_j equals zero or $w(P_i, T_j) = 0$. The formula for determining the state of Petri net after the firing process was expressed in (2).

$$x(k+1) = x(k) + Au \tag{2}$$

Where:
$x(k+1)$ is the token state after firing,
$x(k)$ represents the token state before firing,
A shows the incidence matrix $A = A^+ - A^-$,
u signifies the enabled transition to be fired.

Example 2: Based on Fig. 1, the incidence matrix is obtained

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

Given the initial condition x_0 [3], the liveness classification comprised (a) dead or L_0-*live* when the transition can never be firing sequence for initial condition x_0, (b) L_1-*Live*, if transition can be fired at least once in some firing sequence, (c) L_2-*Live*, for any positive integer k, transition can be fired at least k times in some firing sequence, (d) L_3-*Live*, if transition appears infinitely, often in some firing sequence and (e) L_4-*live*, if transition is $L1$-*live* for every possibility state from x_0. Coverability tree is an approach used to scrutinize the liveness and deadlocks of discrete event systems. Petri net experienced deadlocks when no transitions or sets of transitions could be reinitiated [3] and [1].

C. Max-Plus Algebra

This section describes the basic concepts of max-plus algebra and matrix principles within the context of algebra, accompanied by simple examples. The fundamental principles of this phenomenon can be found in the studies conducted by [4], [2], [5], [6], [11] and [12].

In max-plus algebra $\mathbb{R}_{\max} = R \cup -\infty$, with the two binary operations addition $(\oplus)$ and multiplication $(\otimes)$, are defined as:

$$\begin{aligned} a \oplus b &= \max(a, b)\\ a \otimes b &= a + b \end{aligned}$$

Example 3: $7 \oplus -2 = \max(7, -2) = 7$, $8 \otimes -2 = 8 + (-2) = 6$

Definition 6: The set $\mathbf{R}$ with the two operations $\oplus$ and $\otimes$ is called a max-plus algebra and is denoted by $\mathbf{R}_{\max} = (\mathbb{R}_{\max}, \oplus, \otimes, \epsilon, e)$.

Max-plus algebra featured properties where , $\forall a, b, c \in \mathbb{R}$ satisfies:

$$\begin{aligned} a \oplus (b \oplus c) &= (a \oplus b) \oplus c \qquad a \oplus b = b \oplus a\\ a \otimes (b \otimes c) &= (a \otimes b) \otimes c \qquad a \otimes b = b \otimes a\\ a \otimes (b \oplus c) &= (a \otimes b) \oplus (a \otimes c)\\ a \oplus -\infty &= a \qquad a \oplus a = a\\ a \otimes e &= a, \qquad e = 0 \end{aligned}$$

Similar to classical algebra, the operation $\otimes$ took precedence over the operation $\oplus$.

Theorem 1: The max-plus algebra $\mathbf{R}_{\max} = (\mathbb{R}_{\max}, \oplus, \otimes, \epsilon, e)$ has the algebraic structure of a commutative and idempotent semiring.

Example 4: Based on these properties, the calculations can be performed as follows $7 \otimes -2 \oplus 6 \otimes 3 = (7 \otimes -2) \oplus (6 \otimes 3) = \max(5, 9) = 9$

The structure $(\mathbb{R}_\epsilon, \oplus, \otimes)$ could also be considered as max-plus algebra, and the notation $\mathbb{R}_\epsilon$ was written as $\mathbb{R}_{\max}$. The term of power is defined as follows.:

$$a^{\otimes^n} = \underbrace{a \otimes a \otimes a \otimes \cdots \otimes a}_{\text{n times}} \qquad n \in N$$

for $n = 0$, $a^{\otimes^n} = e$. Hence the power form in algebra max-plus was defined as follows:

$$a^{\otimes^n} = \underbrace{a + a + a + \cdots + a}_{\text{n times}} = n \times a$$

Example 5: $7^{\otimes 2} = 2 \times 7 = 14$; $6^{\otimes -2} = -2 \times 6 = -12$

Operations $(\oplus)$ and $(\otimes)$ could be represented in matrices and vectors similar to classical algebra. If $A, B \in \mathbb{R}_{\max}^{m \times n}$ then

$$\begin{aligned} (A \oplus B)_{ij} &= A_{ij} \oplus B_{ij}, \qquad i = 1, 2, \cdots m, j = 1, 2, \cdot n\\ &= \max\{a_{i,j}, b_{ij}\} \end{aligned} \tag{3}$$

For any matrix $A \in \mathbb{R}_{\max}^{m \times p}$, $B \in \mathbb{R}_{\max}^{p \times n}$, following was obtained matrix $C \in \mathbb{R}_{\max}^{m \times n}$ by the formula $C_{ij} = (A \otimes B)_{ij}$

$$\begin{aligned} C_{ij} &= \bigoplus_{k=1}^{p} A_{ik} \otimes B_{kj}\\ c_{ij} &= \bigoplus_{k=1}^{p} (a_{ik} \otimes b_{kj})\\ &= \max_{k \in p}\{a_{i,k} + b_{k,j}\} \end{aligned} \tag{4}$$

***Example** 6:* Consider $A = \begin{bmatrix} 2 & \epsilon & 5 \\ 1 & 4 & 3 \\ \epsilon & 4 & -1 \end{bmatrix}$ and $B = \begin{bmatrix} -1 & \epsilon & -2 \\ 2 & \epsilon & 3 \\ 4 & 2 & 0 \end{bmatrix}$. Following (4)-(3), then:

$$A \oplus B = \begin{bmatrix} 2 \oplus -1 & \epsilon \oplus \epsilon & 5 \oplus -2 \\ 1 \oplus 2 & 4 \oplus \epsilon & 3 \oplus 3 \\ \epsilon \oplus 4 & 4 \oplus 2 & -1 \oplus 0 \end{bmatrix} = \begin{bmatrix} 2 & \epsilon & 5 \\ 2 & 4 & 3 \\ 4 & 4 & 0 \end{bmatrix}$$

$$A \otimes B = \begin{bmatrix} 1 \oplus \epsilon \oplus 9 & \epsilon \oplus \epsilon \oplus 7 & 0 \oplus \epsilon \oplus 5 \\ 0 \oplus 6 \oplus 7 & \epsilon \oplus \epsilon \oplus 5 & -1 \oplus 7 \oplus 3 \\ \epsilon \oplus 6 \oplus 3 & \epsilon \oplus \epsilon \oplus 1 & \epsilon \oplus 7 \oplus -1 \end{bmatrix} = \begin{bmatrix} 9 & 7 & 5 \\ 7 & 5 & 7 \\ 6 & 1 & 7 \end{bmatrix}$$

For a matrix $A \in \mathbb{R}_{\max}^{m \times n}$, and $\alpha \in R_{\max}$, $\alpha \otimes A$ is defined as follows:

$$(\alpha \otimes A)_{ij} = \alpha \otimes a_{ij}, \forall i \in m, j \in n$$

For a square matrix with positive integer k, matrix $A \in \mathbb{R}_{\max}^{n \times n}$ The k^{th} power of square matrix A is defined recursively by $A^{\otimes 0} = E_n$, for $k = 1, 2, 3 \cdots$

$$A^{\otimes k} = \underbrace{A \otimes A \otimes A \otimes \cdots \otimes A}_{\text{k times}}$$

$$A^{\otimes k} = A \otimes A^{k-1}$$

D. Graph over Max-Plus Algebra

This section describes the graph over max-plus algebra, which can be found in [14] and [5].

According to [14], graph G is is defined as an ordered pair (V, E), where V is a set of elements called vertices, E represents a set of (unordered) pairs of vertices and the elements of E are called edges. A directed graph G is defined as an ordered pair (V, D), where V is a set of vertices and D denotes a set of ordered pairs of vertices. Consequently, the elements of D are called arcs. For each matrix $A \in \mathbb{R}_{\max}^{n \times n}$ a corresponding graph known as the precedence graph of A can be constructed as follows.

***Definition** 7:* (Precedence graph) Consider $A \in \mathbb{R}_{\max}^{n \times n}$. The precedence graph of A, denoted by $G(A)$, is a weighted directed graph with vertices $1, 2, \cdots, n$ and an arc (j, i) with weight a_{ij} for each $a_{ij} \neq \infty$.

***Definition** 8:* (Irreducibility) A matrix $A \in \mathbb{R}_{\max}^{n \times n}$ is considered irreducible if its precedence graph is strongly connected.

Based on Definition 8, in max-plus algebra matrix $A \in \mathbb{R}_{\max}^{n \times n}$ is Irreducible if

$$(A \oplus A^{\otimes 2} \oplus \cdots \oplus A^{\otimes n-1})_{ij} \neq \epsilon, (\forall i, j), i \neq j$$

This condition signifies that, for any two arbitrary vertices i and j of $G(A)$ with $i \neq j$ there exists at least one path (of length $1, 2, 3, \cdots n-1$ from j to i [14] and [18].

***Example** 7:* Consider

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & \epsilon & \epsilon \\ \epsilon & -2 & 2 \end{bmatrix},$$

The precedence graph of A is represented in Fig. 2

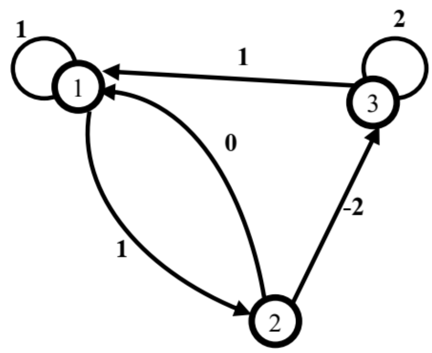

Fig. 2. The precedence graph of the matrix A

III. Main Result

A. Student Tuition Payment Process at the Bank

The processes observed in different banks showed similar flows, even though each service may vary. This study focused on the teller service for student tuition payments and the flow was visually shown in Fig. 3. In this scenario, students arrived at the bank, obtained a number, and joined customer queue. Subsequently, data of student were documented, including name, ID number, payment details, and receipt collection. After the transaction was completed, students exited the queue and the bank.

B. Petri Net Model for Bank Service Student Tuition Fee Payment

After outlining the flow of queue for student tuition payment, Petri net was developed, as shown in Fig. 4. For the student tuition payment process at the bank consisted of 7 places and 10 transitions, as shown in Fig. 4 and Table I. A significant feature of this model was that no places were directly connected to t_1. This indicated that payment processing could occur at any time during the bank's operational hours.

When transition t_1 was fired, a token moved to place P_1, representing the student's entry into the bank. The presence of the token in P_1 enabled transition t_2, continuing the process until transition t_5 was reached. At this point, the token shifted to P_5, enabling transitions t_6 and t_7. When an issue arose during the payment process, firing occurred at transition t_7. Otherwise, if the payment was successful, transition t_6 would be fired. Activation of t_6 moved the token to P_6, enabling transitions t_9 and t_8. In the case of successful payment, t_8 was fired, otherwise, t_9 would be triggered. The process concluded with t_8, moving the token to P_7. This final stage represented the successful completion of the tuition payment process, allowing the student to receive the payment receipt from the bank.

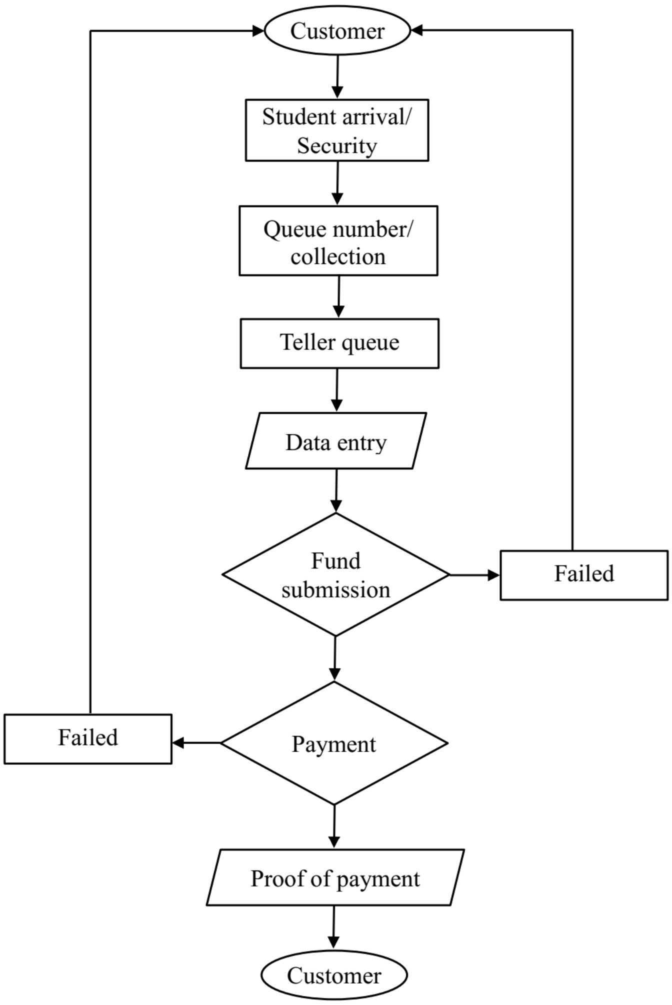

Fig. 3. Flow of bank service for student tuition payment

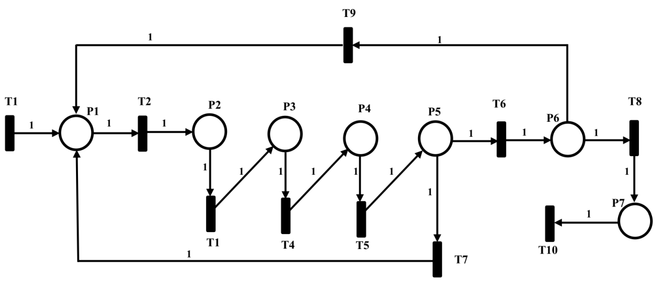

Fig. 4. Petri net design for student tuition payment process

C. Liveness and Deadlock Analysis

The Petri net in Fig. 4 comprised 10 transitions and 7 places, culminating in matrix $A = [a_{ij}]_{7\times 10}$ of the forward matrix included calculating the weights connecting transitions to places, as expressed in (5).

$$A^+ = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5}$$

TABLE I
VARIABLE DEFINITIONS

Variable	Description
P_1	Student enters bank
P_2	Student takes queue number
P_3	Student who queue for payment
P_4	Student provides student data
P_5:	Student completes the transaction with the teller
P_6	Teller payment is completed
P_7	Student receives a payment receipt
t_1	Student arrival process
t_2	Queue number retrieval process
t_3	Queue process
t_4	Student data input process
t_5	Tuition payment submission process
t_6	Teller payment input process
t_7	Failed payment process
t_8	Payment receipt issuance process
t_9	Failed teller payment process
t_{10}	Student leaves bank

The backward matrix calculated the weights connecting places to transitions, as shown in (6).

$$A^- = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Based on (5) and (6), the incidence matrix could be expressed as follows

$$\begin{aligned} A &= A^+ - A^- \\ &= \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \end{aligned} \tag{7}$$

From (2), the possibility of deadlock occurrence within the Petri net was assessed by constructing a coverability tree. The initial state of Petri net shown in Fig. 4 was $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. In this state, t_1 became the enabled transition, and after firing t_1, the subsequent state was established using (2) and (7), as followed:

$$x_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Similarly, if t_2 was fired, Petri net state would result $x_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. The coverability tree that was developed could be seen in Fig. 5.

Based on the liveness classification, it was concluded that the Petri net was deadlock-free for other possible sequences.

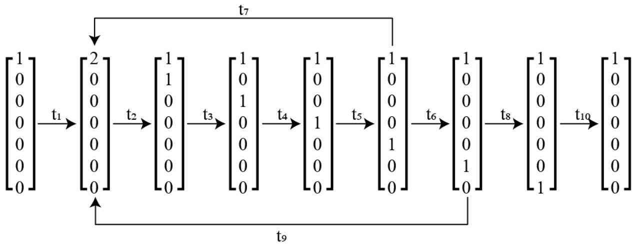

Fig. 5. Coverability tree if t_1 and t_2 is fire

Transition T_0 was *live* since it could always be fired at least once from x_0 and in every state. Transitions t_2 to t_{10} were L_1-*live*, because each could be fired after the preceding transition was fired. The coverability tree model of Petri net was shown in Fig. 5.

D. Max-Plus Algebra Model

Using Petri net model shown in Fig. 4, a max-plus algebra model was constructed to calculate the time required for students to complete tuition payments at the teller. The variables indicating the duration of each stage in the tuition payment process were listed in Tables II and III.

TABLE II
VARIABLES SHOWING TIME

Variable	Description
$t_1(k)$	Student arrival time at k
$t_2(k)$	Student takes queue number at k
$t_3(k)$	Student queue time at k
$t_4(k)$	Student provides data time at k
$t_5(k)$	Student submits payment time at k
$t_6(k)$	Teller payment time at k
$t_7(k)$	Failed payment issuance time at k
t_8(k)	Proof of payment submission time at k
t_9(k)	Failed teller payment time at k
$t_{10}(k)$	Student exits bank time at k

TABLE III
VARIABLES INDICATING PROCESS DURATION

Variable	Description
$v_{t_1,k}$	Duration of the student arrival process
$v_{t_2,k}$	Duration of queue number retrieval
$v_{t_3,k}$	Duration of the student queue
$v_{t_4,k}$	Duration of the student provides data
$v_{t_5,k}$	Duration of the student submits payment
$v_{t_6,k}$	Duration of the teller payment
$v_{t_7,k}$	Duration of failed payment issuance
$v_{t_8,k}$	Duration of proof of payment submission
$v_{t_9,k}$	Duration of failed teller payment
$v_{t_1,k}$	Duration of the student leave bank

According to Petri net flow in Fig. 4 and the variables defined in Tables II and III, the max-plus algebra model was expressed as follows:

$$
\begin{aligned}
t_1(k) =& v_{t_1k} \otimes t_1(k-1) \\
t_2(k) =& v_{t_2k} \otimes (t_1(k) \oplus t_7(k-1) \oplus t_9(k-1))
\end{aligned}
$$

By substituting the equation for $t_1(k)$, the following expression was obtained:

$$
\begin{aligned}
t_2(k) =& v_{t_2k} \otimes (t_1(k) \oplus t_7(k-1) \oplus t_9(k-1)) \\
=& v_{t_2k} \otimes (v_{t_1k} \otimes t_1(k-1) \oplus t_7(k-1) \oplus t_9(k-1)) \\
=& (v_{t_2k} \otimes v_{t_1k}) \otimes t_1(k-1) \\
& \oplus v_{t_2k} \otimes t_7(k-1) \oplus v_{t_2k} \otimes t_9(k-1)
\end{aligned}
$$

Determine $t_3(k)$

$$t_3(k) = v_{t_3k} \otimes t_2(k)$$

By substituting $(t_2(k))$, the following expression was obtained:

$$
\begin{aligned}
t_3(k) =& v_{t_3k} \otimes t_2(k) \\
=& v_{t_3k} \otimes (v_{t_2k} \otimes v_{t_1k} \otimes t_1(k-1) \oplus \\
& v_{t_2k} \otimes t_7(k-1) \oplus v_{t_2k} \otimes t_9(k-1)) \\
=& (v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes t_1(k-1) \\
& \oplus (v_{t_3k} \otimes v_{t_2k}) \otimes t_7(k-1) \oplus \\
& (v_{t_3k} \otimes v_{t_2k}) \otimes t_9(k-1)
\end{aligned}
$$

Determine $t_4(k)$

$$t_4(k) = v_{t_4k} \otimes t_3(k)$$

By substituting equation $t_3(k)$, the following expression was obtained:

$$
\begin{aligned}
t_4(k) =& v_{t_4k} \otimes t_3(k) \\
=& v_{t_4k} \otimes ((v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes t_1(k-1) \oplus \\
& (v_{t_3k} \otimes v_{t_2k}) \otimes t_7(k-1) \oplus (v_{t_3k} \otimes v_{t_2k}) \otimes t_9(k-1)) \\
=& (v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes t_1(k-1) \\
& \oplus (v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_7(k-1) \oplus \\
& (v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_9(k-1)
\end{aligned}
$$

In a similar manner, the equations for $t_5(k), t_6(k), t_7(k), t_8(k), t_9(k)$ and $t_{10}(k)$ are determined.

$$
\begin{aligned}
t_5(k) =& (v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes t_1(k-1) \\
& \oplus (v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_7(k-1) \oplus \\
& (v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_9(k-1)
\end{aligned}
$$

$$
\begin{aligned}
t_6(k) =& (v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes t_1(k-1) \\
& \oplus (v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_7(k-1) \oplus \\
& (v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_9(k-1)
\end{aligned}
$$

$$
\begin{aligned}
t_7(k) =& (v_{t_7k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes t_1(k-1) \\
& \oplus (v_{t_7k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_7(k-1) \oplus \\
& (v_{t_7k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_9(k-1)
\end{aligned}
$$

$$
\begin{aligned}
t_8(k) =& (v_{t_8k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes \\
& t_1(k-1) \oplus (v_{t_8k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes \\
& t_7(k-1) \oplus (v_{t_8k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \\
& \otimes t_9(k-1)
\end{aligned}
$$

$$
\begin{aligned}
t_9(k) =& (v_{t_9k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes \\
& t_1(k-1) \oplus (v_{t_9k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \\
& \otimes t_7(k-1) \oplus (v_{t_9k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \\
& \otimes t_9(k-1)
\end{aligned}
$$

$$t_{10}(k) = (v_{t_{10}k} \otimes v_{t_8k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k}) \otimes t_1(k-1) \oplus (v_{t_{10}k} \otimes v_{t_8k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_7(k-1) \oplus (v_{t_{10}k} \otimes v_{t_8k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k}) \otimes t_9(k-1)$$

The synchronized model from $t_1(k)$ to $t_{10}(k)$ was then represented in matrix form as shown in (8)

$$\begin{bmatrix} t_1(k) \\ t_7(k) \\ t_9(k) \\ t_{10}(k) \end{bmatrix} = \begin{bmatrix} v_{t_1k} & \epsilon & \epsilon & \epsilon \\ b & c & c & \epsilon \\ d & e & e & \epsilon \\ f & g & g & \epsilon \end{bmatrix} \otimes \begin{bmatrix} t_1(k-1) \\ t_7(k-1) \\ t_9(k-1) \\ t_{10}(k-1) \end{bmatrix} \tag{8}$$

where,

$$\begin{aligned} b =& v_{t_7k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \otimes v_{t_1k} \\ c =& v_{t_7k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \\ d =& v_{t_9k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes \\ & v_{t_2k} \otimes v_{t_1k} \\ e =& v_{t_9k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \\ f =& v_{t_{10}k} \otimes v_{t_8k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes \\ & v_{t_2k} \otimes v_{t_1k} \\ g =& v_{t_{10}k} \otimes v_{t_8k} \otimes v_{t_6k} \otimes v_{t_5k} \otimes v_{t_4k} \otimes v_{t_3k} \otimes v_{t_2k} \end{aligned} \tag{9}$$

This model enabled the analysis and simulation of the time required for the student payment process using the principles of max-plus algebra.

IV. Numerical Experiments

In this section, the results of the numerical experiments were presented. When the duration of each variable was provided in seconds as follows: $v_{t_11} = 30, v_{t_21} = 10, v_{t_31} = 3469, v_{t_4} = 15, v_{t_51} = 15, v_{t_61} = 25, v_{t_71} = 20, v_{t_81} = 10, v_{t_91} = 20, v_{t_{10}1} = 60$ the values for the coefficients in (9) were calculated as $b = 3559, c = 3529, d = 3584, e = 3554, f = 3634$ and $g = 3604$. Given these initial conditions and using the corresponding equations, the following state was obtained.

$$\begin{bmatrix} t_1(0) \\ t_7(0) \\ t_9(0) \\ t_{10}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

After performing the calculations, the next state was expressed as follows

$$\begin{bmatrix} t_1(1) \\ t_7(1) \\ t_9(1) \\ t_{10}(1) \end{bmatrix} = \begin{bmatrix} 30 & \epsilon & \epsilon & \epsilon \\ 3559 & 3529 & 3529 & \epsilon \\ 3584 & 3554 & 3554 & \epsilon \\ 3634 & 3604 & 3604 & \epsilon \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 30 & 3559 & 3584 & 3634 \end{bmatrix}^T$$

This result signified that the total time for a student to complete the tuition payment process, from arrival at the bank to departure, was 3634 seconds.

V. Conclusion

In conclusion, this study introduced Petri net and max-plus algebra to model the queue system for student's tuition fee payments in banks. The Petri net model, which represented the student tuition payment service, comprised 7 places and 10 transitions, forming a matrix of size $(A_{7\times10})$. Simultaneously, Max-plus algebra model was formulated, as shown in (8),enabling the analysis of the duration of the payment process. The results showed that the total duration of the payment process, from the student's arrival to their departure, was 3634 seconds, or approximately 1 hour and 34 seconds. This finding provided valuable insights into the bank's queue dynamics, which could be used to optimize service delivery. By understanding service time for each student, banks could implement measures to reduce wait times.

References

[1] A. Sadiq, F. Ahmad, S. A. Khan, J. C. Valverde, T. Naz, and M. W. Anwar, "Modeling and analysis of departure routine in air traffic control based on Petri nets" *Neural Computing and Applications*, vol. 25, pp. 1099–1109, 2014.

[2] R. Cuninghame-Green, "Minimax Algebra, Lecture Notes in Economics and Mathematical Systems", *Springer-Verlag*, 1979.

[3] C. G. Cassandras, S. Lafortune, "Introduction to Discrete Event Systems: Second Edition" *Springer*, 2008.

[4] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat, "Synchronization and Linearity: An Algebra for Discrete Event Systems" *New York: John Wiley and Sons,* 1992.

[5] B. Heidergott, G. Olsder, and J. van der Woude, "Max Plus at Work". *Princeton, New Jersey: Princeton University Press*, 2006.

[6] Subiono, "On Classes of Min-Max-Plus Systems and Their Applications", *PhD thesis, Delft University of Technology, The Netherlands*, 2000.

[7] M. H. Andersen, "Max-Plus Algebra: Properties and Applications", *Master of Science in Mathematic Thesis Department of Mathematics, Laramie, WY*, 2002.

[8] J. G. Braker, "Algorithms and Applications in Timed Discrete Event Systems", *PhD thesis, Delft University of Technology, The Netherlands*, 1993.

[9] N. K. Krivulin, "Using Max-Algebra Linear Models in the Representation of Queueing Systems", *Proceedings of the 5th SIAM Conference on Applied Linear Algebra*, Snowbird, UT, SIAM, Philadelphia, pp. 155-160, 1994.

[10] N. K. Krivulin, "A Max-Algebra Approach to Modeling and Simulation of Tandem Queueing Systems", *Mathematical and Computer Modelling*, vol. 22, pp. 25-31, 1995.

[11] Z. R. Konigsberg, "Timed Petri nets modeling and Lyapunov/max-plus algebra stability analysis for a type of queuing systems", *International Journal of Pure and Applied Mathematics*, vol. 86, pp. 301-323, 2013.

[12] B. D. Schutter and T. v. d. Boom, "Max-plus algebra and max-plus linear discrete event systems: An introduction". *9th International Workshop on Discrete Event Systems, WODES'08*, pp. 36–42, 2008.

[13] H. A. Bermanei, J. M. Böling G. Högnäs, "Modeling and scheduling of production systems by using max-plus algebra", *Flexible Services and Manufacturing Journal*, vol. 36, pp. 129–150, 2024.

[14] B. D. Schutter, "Max-Algebraic System Theory for Discrete Events Systems", *Katholieke Universiteit Leuven, Belgium*, 1996.

[15] N. Nurwan, S. Ismail, M.R.F. Payu, L. Yahya, and D. Wungguli, "Input-Output Analysis on Pia Saronde Production Process Scheduling with Invariant Max-Plus Linear System", *KnE Life Sciences: International Conference On Mathematics And Science Education*, pp. 166-176, 2024.

[16] J. Zhou, and G. Reniers, "Petri-net based modeling and queuing analysis for resource-oriented cooperation of emergency response actions", *Journal of Loss Prevention in the Process Industries: the international journal of chemical and process plant safety,*, pp. 554-562, 2016.

[17] G. A. Martha, S. Siswanto, and T. A. Kusmayadi, "Petri Net and Max-Plus Algebra Model in Bank Queue with Two Servers in Each Service", *Journal of Physics: Conference Serie*, 2019.

[18] B. D. Schutter, T. v. d. Boom, J. Xu and · S. S. Farahani, "Analysis and control of max-plus linear discrete-eventsystems: An introduction", *Discrete Event Dynamic Systems* , vol. 30, pp. 25-54, 2020.

Nurwan Nurwan is Assistant Professor of Department of Mathematics, Universitas Negeri Gorontalo, Indonesia, since September 2010. His research area include max-plus algebra and operational research. His education background are as follows: (1) M.S. degree from Department of Mathematics, faculty of Mathematics and Natural Science, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia, 2008, majoring in max-plus algebra; (2) B.S. degree from Department of Mathematics, faculty of Mathematics and Natural Science, Universitas Negeri Gorontalo, Gorontalo, Indonesia, 2004. He is a member of IAENG Society of Computer Science and IAENG Society of Operations Research, and member of the Indonesia Mathematical Society (INDO-MS).

Stenly Pranata was born in Palu Central Sulawesi, Indonesia in 2001. He is undergraduate mathematics student from Universitas Negeri Gorontalo in 2024. His major field of study was max-plus algebra.

Isran K. Hasan was born in Gorontalo, Indonesia in 1990. He received an M.S. degree in Mathematics from Bandung Institute of Technology in 2015. His major field of study was Applied Statistics and Probability. Since 2017, he has been with the Department of Statistics, Universitas Negeri Gorontalo. Isran is a member of the Indonesia Mathematical Society (INDO-MS) and the Indonesian Statistics Higher Education Forum (FORSTAT). He has been an editor and Reviewer of many Indonesian journals.