BAREKENG: Journal of Mathematics and Its Applications
March 2025 Volume 19 Issue 1 Page 0303–0318
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19iss1pp0303-0318

SOLVING CUTTING STOCK PROBLEM USING PATTERN GENERATION METHOD ON 2-DIMENSIONAL STOCK

Ihda Hasbiyati [1*], Indri Latifa[2] , Abdul Rahman[3], Ahriyati[4], Moh Danil Hendry Gamal[5]

[1,2,5]Department of Mathematics, Faculty of Mathematics and Science, Universitas Riau
Kampus Bina Widya KM. 12,5, Simpang Baru, Tampan Subdistrict, Pekanbaru, 28293, Indonesia
[3]Department of Mechatronics, Faculty of Engineering, Universitas Lancang Kuning
Jln. Yos Sudarso KM.8, Umban Sari, Pekanbaru, 28265, Indonesia
[4]Department of Mathematics, Faculty of Mathematics and Science, Universitas Palangkaraya
Jln. Yos Sudarso, Palangkaraya, 73111, Indonesia

*Corresponding author's e-mail: * **ihdahasbiyati@lecturer.unri.ac.id***

Article History:
Received: 2nd June 2024
Revised: 29th November 2024
Accepted: 29th November 2024
Published: 13th January 2025

Keywords:
Stock Cutting Problem;
2-Dimensional Stock;
Pattern Generation Method;
Simplex Method;
Linear Programming.

ABSTRACT

This article discusses the solution of the 2-dimensional stock cutting problem using the Branch and Bound modified pattern generation method. The pattern generation method will produce a feasible cutting pattern matrix which is then converted into a mathematical model with a linear program equation with an objective function to minimize the use of initial stock materials. The research is a case study located at the Handal Karya Buana Store which is engaged in cutting glass of different sizes, thicknesses and types of glass. In this case, 3 types of initial stock will be used with the same thickness, and type but have different area sizes, and one of the consumer demand data will be used, namely 3 types of requests with different sizes and many requests. By using the pattern generator method, 10 cutting patterns are generated with each different cutting residue. By using the simplex method, the optimal solution is obtained for the amount of initial stock needed, the pattern used and the remaining cuts produced. So using the pattern generator method can produce a feasible cutting pattern, and can be used as an alternative to solve the stock cutting problem.

How to cite this article:
I. Hasbiyati and I. Latifa., "SOLVING CUTTING STOCK PROBLEM USING PATERN GENERATION METHOD ON 2-DIMENSIONAL STOCK," *BAREKENG: J. Math. & App.*, vol. 19, iss. 1, pp. 0303-0318, March, 2025.

1. INTRODUCTION

One of the most common problems in industries such as steel plate, glass, wood board, cardboard, and paper is the problem of cutting 2-dimensional or rectangular stock. In the 2-dimensional stock cutting problem, the initial stock will be cut into smaller rectangular pieces of a certain size called units. The initial stock must be optimally cut to be able to meet the many requests for each unit, so planning is needed in determining the right stock cutting pattern so as to minimize the use of stock or minimize the remaining stock cuts.

According to Gilmore and Gomory (1961) [1], the stock cutting problem is the problem of fulfilling demand by minimizing the expenditure for a given amount of material width to be cut and of a given stock length and at a given cost. When the stock-cutting problem is expressed in integer programming form, the many variables involved generally make computation difficult. The same difficulty also occurs when only approximate solutions are sought with linear programs, so Gilmore and Gomory introduced one of the solution methods for this stock cutting problem, namely the column generation method [1].

Heassler and Sweeney [2] introduced the types of dimensions in the stock cutting problem including 1-dimensional, 1.5-dimensional, rectangular or 2-dimensional. Heassler and Sweeney discussed some basic formulation problems and solution procedures for solving stock cutting problems namely, linear programming, sequential heuristics, and hybrid solution procedures. As for 2-dimensional stocks, Heassler and Sweeney recommend taking an approach to solving large problems with constraints on how many ordered sizes can appear in a pattern.

Octarina et al. [3] implemented a pattern generation algorithm to formulate the Gilmore and Gomory model of two-dimensional CSP. The constraints of the Gilmore and Gomory model were performed to ensure that the strips cut in the first stage will be used in the second stage. Branch and Cut method was used to obtain the optimal solution.

Discussions on 2-dimensional stock cutting have been conducted by several researchers. Slimi and Abbas [4] discussed solving the 2-dimensional stock cutting problem with the dual objective of minimizing the remaining cuts and the amount of stock used. The discussion is about cutting a number of rectangular pieces from a set of identical new material plates. Slimi and Abbas minimize the residual cutting stock and manage the amount of stock usage required. The technique consists of two stages, the first one generates all possible cutting patterns and the second one allows to create of a cutting plan, meeting the demand.

Erdem and Kasimbeyli [5] apply a two-step mathematical programming-based heuristic solution approach to the two-dimensional guillotine cutting stock problem. In the first step, all products that need to be cut from the stock are considered without regard to placement constraints. In the second step, a mathematical model is constructed to generate cutting patterns for the demand list produced in the first step, using the appropriate stock material, while taking length and width constraints, as well as other relevant assumptions, into account. In addition, Octarina et al. [6] solved a 2-dimensional stock cutting problem with a pattern generation method, which was then transformed into the form of Gilmore and Gomory models. Furthermore, Dodge et al. [7] solved the 2-dimensional stock cutting problem with a DNA algorithm.

Atika et al. [8] addressed the 3-dimensional stock cutting problem of a large block being cut into smaller blocks, each with a certain size and number of requests. Their research aims to modify the pattern generation algorithm so that it can be used in 3-dimensional problems and can find a cutting pattern with the minimum remaining cuts. The blocks will be cut based on length, width, and height, then the remaining cuts will be re-cut if possible to fulfill other request sizes. In addition, Atika et al also considered the use of orthogonal cutting or cutting by rotating a certain number of degrees so that the cutting dimensions would have six permutations.

Caricato and Grieco [9] also conducted research on the application of stock cutting problems to production planning in film packaging. Production planning problems that are often discussed include contrasting goals and strategies between customers and production optimization. Then, Rahman et al. [10] conducted research on the solution to the 2-dimensional stock cutting problem using the column generation method approximated by dynamic programming.

Rodrigo et al. [11] conducted another research on 1-dimensional stock using the help of the phyton program and the cartesian coordinate system. Furthermore, Bangun et al. [6] conducted research on the implementation of the stock cutting problem with the Branch and Cut method on the 2-dimensional stock with the N-sheet model. Based on the results of his research, the N-sheet model ensures the possibility of trim loss is minimized to meet demand.

Lomate et al. [12] discussed the stock cutting problem which is decomposed into two parts, namely pattern generation and optimization. Pattern generation is done by greedy optimization and optimization is done by integer programming. The stock dimensions used are 1-dimensional stock and 2-dimensional stock. And Alten et al. [13], conducted case study research using 3-dimensional stock in the case of mattress production using the column generation method.

In this article, researchers discuss solving the 2-dimensional stock cutting problem using the pattern generation method and its application to the real world, namely in the glass field. The data source is obtained from Handal Karya Buana glass shop located in Pekanbaru. Then the data is solved by the simplex method using the help of Microsoft Excel's Solver Table software.

2. RESEARCH METHODS

2.1 Linear Programming

Hillier and Leiberman [14] state that linear programming uses a mathematical model to describe the problem under discussion. The word linear means that all mathematical functions expressed in this model are linear functions. Then, the word programming means planning.

Thus, linear programming is a plan that uses linear functions to get optimal results. Linear programming aims to obtain demand-based desires such as maximizing profits or minimizing production costs from a problem. The problems that use linear programming are economic, industrial, social, military, and others.

Before using linear programming, here are some basic components that form the mathematical model of linear equations [15]:

a. Desicion variables

   In making a mathematical model, it is necessary to first define the decision variables that are relevant to the elements of the problem.

b. Objective function

   In linear programming problems, decision-making on maximization (generally profit) or minimization (generally cost) of decision variables is referred to as an objective function.

c. Constraints

   Constraint function is a function as a condition of constraints that must be met or become a standard for solving mathematical models.

d. Sign restrictions

   To complete the general form of linear programming, a sign restriction is required for each decision variable. If the decision variable is assumed to be non-negative then $x_i \geq 0$, if it is negative then $x_i < 0$. If the variable x_i is assumed to be positive or negative (or equal to 0), then x_i is called unconstrained by the sign.

Then form a mathematical model by determining the objective function and some constraints of the problem. The general form of the linear programming model [14] is as follows:

$$max\ z = c_1x_1 + c_2x_2 + \cdots + c_nx_n, \tag{1}$$

constrains

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1, \\ a_{21}x_2 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2, \\ \vdots \qquad \vdots \quad &\vdots \ \vdots \\ a_{m1}x_m + a_{m2}x_m + \cdots + a_{mn}x_n &\leq b_m, \end{aligned} \tag{2}$$

and

$$x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0. \tag{3}$$

Descriptions:

z : objective function (total optimized function value),

x_j : decision variable j^{th} $(j = 1, 2, \dots, n)$,

c_j : decision variable coefficient j^{th} ,

b_i : constraint function value i^{th} $(i = 1, 2, \dots, m)$,

$a_{i,j}$: the coefficient of the j^{th} decision variable of the i^{th} constraint function.

Equation (1) can be referred to as the objective function to maximize. **Equation (2)** is called the constraint function. Similarly, **Equation (3)** is called a non-negative constraint function.

The stages in solving linear programming optimization include determining the decision variables, then forming the objective function and formulating the constraints and solving the problem using the graph approach method or the simplex method. The graph method is used to solve linear programming problems if the variables in the problem are no more than two variables. In comparison, programming problems with more than two variables are solved by the simplex method.

2.2 Simplex Method

George Dantzig developed an efficient method for solving linear programming problems, namely the simplex algorithm **[15]**. The development of this algorithm has been widely used to solve problems in diverse industries. Industrial problems such as banking, education, forestry, petroleum, and transportation.

The simplex method is a method with basic concepts of geometry using algebraic procedures **[14]**. Understanding geometry concepts can help in operating the simplex method. Here are some terms in the simplex method that need to be known:

a. Optimum solution

An ptimum solution is a feasible solution that optimizes a linear programming problem.

b. Feasible regions

The feasible region is the set of numbers that contain feasible solutions.

c. Iteration

Iteration is a calculation stage in the form of a table with the value of the next table depending on the value of the previous table.

d. Basis variable vector

The basis variable vector is a column vector or matrix of size $m \times 1$ that has a non-zero value in any iteration table.

$$x_{BV} = \left[x_{BV_1} \; x_{BV_2} \; \vdots \; x_{BV_m} \right]$$

e. Nonbasis variable vector

The nonbasis variable vector is a column vector or matrix of size $(n - m) \times 1$ that is zero at any iteration table.

$$x_{NBV} = \left[x_{NBV_1} \; x_{NBV_2} \; \vdots \; x_{NBV_{(n-m)}} \right]$$

f. Basis variable coefficient

The coefficient of the basis variable is the objective function coefficient for the basis variable in the form of a $1 \times m$ matrix.

$$c_{BV} = \left[c_{BV_1} \; c_{BV_2} \; \cdots \; c_{BV_m} \right]$$

g. Nonbasis variable coefficients

The nonbasis variable coefficient is the objective function coefficient for the base variable in the form of a $1 \times (n - m)$ matrix.

$$c_{NBV} = \left[c_{NBV_1} \; c_{NBV_2} \; \cdots \; c_{NBV_{(n-m)}} \right].$$

h. Slack variable

The slack variable is a variable added to the constraint function in the mathematical model to convert the inequality ($\leq$) into an equation ($=$).

i. Surplus variable

Surplus variables are variables added to the constraint function in the mathematical model to convert inequality ($\geq$) into an equation ($=$).

j. Artificial variable

Artificial variables are variables that are added to the constraint function because the constraint function does not contain base variables.

k. Entry and exit variables

The entry variable is the variable chosen to be the base variable in the next iteration. While the out variable is a variable that replaces the incoming variable in the next iteration.

l. Pivot column

The pivot column is the column that contains the entry variable.

m. Pivot row

The pivot row is the row that contains the outgoing variable.

The first step in solving the simplex method is to convert the linear programming problem into standard form. This linear program problem can be converted into an equivalent problem with all of its constraints in the form of equations and all variables are non-negative [16]. Several things that need to be considered in the simplex method, namely as follows:

a. The constraint function with inequality sign ($\leq$) in general form is converted into equation ($=$) then added slack variables,

b. The constraint function with inequality sign ($\geq$) in general form is converted into equation ($=$) and then subtracted from the surplus variable,

c. The constraint function with equality sign ($=$) in the general form are added artificial variables.

The general form of linear programming in **Equation (1)** - **Equation (3)**, is converted into the standard form of the simplex method:

$$max\ z = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

constraints

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + s_1 &= b_1, \\ a_{21}x_2 + a_{22}x_2 + \cdots + a_{2n}x_n + s_2 &= b_2, \\ \vdots \quad &= \vdots \\ a_{m1}x_m + a_{m2}x_m + \cdots + a_{mn}x_n + s_m &= b_m, \end{aligned}$$

and

$$x_1, x_2, \dots, x_n, s_1, s_2, \dots, s_m \geq 0.$$

Descriptions:

z := objective function (optimized function value),

x_j := decision variable j^{th},

c_j := decision variable coefficient of the objective function j^{th},

b_i := constraint function value i^{th},

$a_{i,j}$:= decision variable coefficient j^{th} of constraint function i^{th},

s_i := slack variable in constraint i^{th}.

The standard form of the simplex method above can be simplified into matrix form as follows [17]:

$$max \quad \sum_{j=1}^{n} c_j x_j,$$

$$constraint \quad \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m,$$

$$x_j \geq 0, \quad j = 1,2, \dots, n.$$

Then denote the row vectors $(c_1, c_2, \dots, c_n)$ as c, and the column vectors x and b also the matrix A is $m \times n$.

$$x = [x_1\ x_2\ \vdots\ x_n\ ], \quad b = [b_1\ b_2\ \vdots\ b_n\ ], \quad A = [a_{11}\ a_{12}\ a_{21}\ a_{22}\ \cdots\ \cdots\ a_{1m}\ a_{2m}\ \vdots\ \vdots \ddots \vdots\ a_{n1}\ a_{n2}\ \cdots\ a_{nm}\ ].$$

Furthermore, it can be rewritten in matrix form. The linear programming matrix notation is as follows [17]:

$$max \quad cx,$$

$$constraint \quad Ax = b,$$

$$x \geq 0. \tag{4}$$

Suppose matrix A is expressed as $A = [D \quad B]$ where D is an $m \times (n-m)$ submatrix of A containing the coefficients of the nonbasis variables,

$$D = [a_{11}\ a_{12}\ a_{21}\ a_{22}\ \cdots\ \cdots\ a_{1(n-m)}\ a_{2(n-m)}\ \vdots\ \vdots\ \vdots\ a_{m1}\ a_{m2}\ \cdots\ a_{m(n-m)}\ ],$$

while B is an $m \times m$ submatrix of A that contains the coefficients of the base and linear base variables,

$$B = [a_{1(n-m+1)}\ a_{1(n-m+2)}\ a_{2(n-m+1)}\ a_{2(n-m+2)}\ \cdots\ \cdots\ a_{1n}\ a_{2n}\ \vdots\ \vdots\ \vdots\ a_{m(n-m+1)}\ a_{m(n-m+2)}\ \cdots\ a_{mn}\ ],$$

Since B is a singular matrix, B has an inverse which can be formalized as follows:

$$Bx_{BV} = b,$$

suppose x_{BV} is a vector of basis variables $x_{BV} = [x_{BV1}\ x_{BV2}\ \cdots\ x_{BVm}\ ]^T$ so that

$$x_{BV} = B^{-1}b. \tag{5}$$

Then the variable x in **Equation (4)** which is an $n \times 1$ matrix consisting of the first m components is the basis variable x_{BV} and the remaining nonbasis variable x_{NBV} is 0. x can be expressed as $x = [x_{BV} \quad 0]^T$, so x is the solution to the constraint function $Ax = b$. Furthermore, the value of z can be calculated using:

$$z = c_{BV} x_{BV} = c_{BV} B^{-1} b, \tag{6}$$

where c_{BV} is the objective function coefficient and corresponds to the basis variable x_{BV} [16].

Suppose the standard form has a constraint function of m equations with n decision variables ($m \leq n$) containing nonnegative variables so that there are $n - m$ nonbasis variables. If the base variable of a base solution is zero then the base solution is called a degenerate base solution. A vector x that satisfies $Ax = b$ is said to be a feasible solution when $x \geq 0$.

The basis solution in the simplex method replaces the extreme points on the feasible region graph so that the maximum number of iterations of the simplex method is equal to the maximum number of basis solutions in standard form. Therefore the number of iterations in the simplex method is no more than [16].

$$C_m^n = \frac{n!}{m!\,(n-m)!}$$

The stages of solving linear programming problems for minimization using the simplex method with tables [15] are:

a. Change the form of linear equations in standard form

Form the mathematical model for the minimization problem as follows:

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \, , \tag{7}$$

constraint

$$\begin{array}{lllllllllll} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & + & s_1 & & & = & b_1, \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{1n}x_n & & & + & s_2 & = & b_2, \\ & & & & & & \vdots & & & & & = & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & & & + & s_m & = & b_m, \end{array} \tag{8}$$

$$x_j, s_i \geq 0, \quad for \quad i = 1, 2, \ldots, m, \quad j = 1,2, \ldots, n. \tag{9}$$

b. Determining the base feasible solution and forming the initial simplex table

Slack variables can be used as base variables for constraints if the right-hand side of the constraint has a nonnegative value [16]. Equation (7), Equation (8), and Equation (9) can be transformed into the form of a simplex table, as follows:

Table 1. Simplex Table

Row	Basis	z	x_1	$\cdots$	x_n	s_1	$\cdots$	s_m	RHS	Ratio
0	z	0	$-c_1$	$\cdots$	$-c_n$	0	$\cdots$	0	z	-
1	s_1	0	a_{11}	$\cdots$	a_{1n}	1	$\cdots$	0	b_1	R_1
2	s_2	0	a_{21}	$\cdots$	a_{2n}	0	$\cdots$	0	b_2	R_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
m	s_m	0	a_{m1}	$\cdots$	a_{mn}	0	$\cdots$	1	b_m	R_m

In Table 1, Row 0 shows the objective function. Rows $1, 2, \ldots, m$ denotes the constraint function. Column RHS represents the value of the right-hand side (b_i) on the constraint function. The Ratio column is a ratio test of the right-hand side values with certain variable coefficients.

c. Perform optimization test to determine the pivot column

Check the presence of positive elements in Row 0. If all nonbasis variables have negative coefficients, then the base feasible solution is optimal. Otherwise, if there is a nonnegative coefficient in Row 0, select the nonbasis variable with the largest coefficient value in Row 0 (select any if there is more than one). Then, the nonbasis variable is swapped to the base variable or called the entry variable.

d. Determining the pivot row

Check the presence of positive elements in the pivot column below Row 0. If all coefficients are already negative in the pivot column this indicates that there is no finite optimal solution. On the other hand, if there are nonnegative coefficients in the pivot column, then perform a ratio test and select the element with the largest ratio (select any if there is more than one).

e. Form a new table by pivoting (iteration) and repeat step (ii).

Definition 2.1 [18] Any minimum-valued vector x in the objective function that satisfies the constraint $Ax = b$, with $x \geq 0$, can be said to be an optimal feasible solution. An optimal feasible solution that contains a basis is called an optimal basis feasible solution.

2.3 Cutting Stock Problem

According to Gilmore and Gomory [1], the cutting stock problem commonly abbreviated as CSP is the problem of fulfilling demand at minimum cost from a length of stock that is cut into a certain length of material at a certain cost. The goal to be achieved from the cutting stock problem is to minimize the use of initial stock.

The general form for the cutting stock problem with the objective function of minimizing initial stock usage is as follows:

$$z = \sum_{j=1}^{n} x_j,$$

constraint

$$\sum_{j=1}^{n} p_{ij} x_j \geq d_i,$$

$$x_j \geq 0 \ \ and\ integer.$$

Descriptions:

z : Objective function (optimized objective function value),

x_j : Number of starting stocks required to cut with the j^{th} pattern ($j = 1,2, \dots, m$),

p_{ij} : The j^{th} pattern cut with the i^{th} demand size,

d_i : Number of i^{th} requests ($i = 1,2, \dots, n$).

2.4 Simplex Method

According to Octarina [6] the pattern generation method is one of the algorithms to solve the pattern cutting problem. The stock problem used is 2-dimensional stock, so the stock size will be $L \times W$ dimension. The pattern cutting obtained using the pattern generation method will be cut into units of length (l) and width (w) with a certain number of requests.

The idea developed in this method is that a matrix P of size $n \times m$ will be formed which contains the feasible cutting patterns used by the column vector P_j with j is the pattern generated from this method, then P_j consists of p_{ij} obtained from the calculation using the pattern generation method with i as many as the types of stock unit requests ($i = 1, 2, \dots, n$). In addition, there are also many requests for each unit or demand called d_i. P can be written as follows:

$$P = [P_1\ P_2\ \cdots\ P_m\ ] = [p_{11}\ p_{12}\ p_{21}\ p_{22}\ \cdots \cdots\ p_{1m}\ p_{2m}\ \vdots\ \vdots\ \ddots\ \vdots\ p_{n1}\ p_{n2}\ \cdots\ p_{nm}\ ] \tag{10}$$

In the 2-dimensional stock cutting problem, consider that the stock cutting problem consists of cutting a rectangular stock with length L and width W into pieces called chunks with length $l_i \leq L$ and width $w_i \leq W$ with $i = 1, 2, \dots, n$, the rectangles in the stock are available in unlimited number and $L \times W$ is of the same size [4].

The following is the algorithm of the pattern generation method with modified Branch and Bound [15]:

Step 1. Sort the length (l_i) in the order of longest to shortest, $l_1 > l_2 > \cdots > l_n$. Then arrange the width (w_i) based on the length (l_i).

Step 2. For $i = 1, 2, \dots, n$ form the first pattern P_1 of P from Step 3 - 5.

Step 3. Calculate a_{ij} the number of cutting patterns of demand units based on the initial stock length with i being the i^{th} demand unit and j being the j^{th} cutting pattern. a_{ij} is calculated using the floor operation $\lfloor\ \rfloor$ because a cutting pattern can be cut if the demand size can fit into the initial stock length. Set $i = 1, j = 1$ to start the formation of the initial pattern, a_{11} ie,

$$a_{11} = \lfloor \frac{L}{l_1} \rfloor. \tag{11}$$

while to calculate a_{ij} is done as follows:

$$a_{ij} = \lfloor \frac{L - \sum_{z=1}^{i-1} a_{zj} l_z}{l_i} \rfloor. \tag{12}$$

Step 4. Calculate b_{ij} i.e. the number of demand unit cutting patterns generated from a_{ij} based on the initial stock width. If $a_{ij} \geq 0$ then b_{ij} can be calculated. b_{ij} is calculated using the floor operation $\lfloor\ \rfloor$ because the cutting pattern can be cut if the request size can fit into the initial stock width.

$$b_{ij} = \{\lfloor \frac{W}{w_i} \rfloor, \ for\ a_{ij} > 0.0, \quad else. \tag{13}$$

Step 5. Calculate the value of P_1 with $p_{ij} = a_{ij}b_{ij}$.

Step 6. Calculate the cutting residue

a. Remaining cut based on stock length (c_u)

$$c_u = \left(L - \sum_{i=1}^{n} a_{ij}l_i\right) \times W \tag{14}$$

For $i = 1, 2, \ldots, n$, if $\left(L - \sum_{i=1}^{n} a_{ij}l_i\right) \geq w_i$ and $W \geq l_i$, then a new pattern cut can be formed by rotating by 90° to the size of the i^{th} demand unit. The new pattern cut can be calculated as follows:

$$A_{ij} = \lfloor \frac{L - \sum_{i=1}^{n} a_{ij}l_i}{w_i} \rfloor, \ B_{ij} = \{\lfloor \frac{W}{l_i} \rfloor, \ for\ a_{ij} > 0.0, \quad else. \tag{15}$$

and $P_j = P_j + A_{ij}B_{ij}$. As for others,

$$A_{ij} = 0,$$

$$B_{ij} = 0,$$

$$P_j = P_j$$

Next, for $A_{ij} > 0$ then calculate the remaining cutting as follows:

$$c_u = \left(\left(L - \sum_{i=1}^{n} a_{ij}l_i\right) - A_{ij}w_i\right) \times B_{ij}l_i$$

$$c_v = \left(L - \sum_{i=1}^{n} a_{ij}l_i\right) \times (W - B_{ij}l_i) \tag{16}$$

while for others, calculate with **Equation (14)**.

b. Remaining cut based on stock width (c_v)

$$c_v = \left(a_{ij}l_i\right) \times k_{ij} \tag{17}$$

with $k_{ij} = W - (b_{ij}w_i)$, but if $b_{ij}w_i = 0$ then $k_{ij} = 0$. With $z \neq i$ if $\left(a_{ij}l_i\right) \geq l_z$ and $k_{ij} \geq w_z$, then a new cutting pattern can be formed as follows:

$$A_{zj} = \lfloor \frac{L - \sum_{i=1}^{n} a_{ij}l_i}{w_i} \rfloor, \ B_{ij} = \{\lfloor \frac{W}{l_i} \rfloor, \ for\ a_{ij} > 0.0, \quad else. \tag{18}$$

and $P_j = P_j + A_{ij}B_{ij}$. As for others,

$$A_{ij} = 0,$$

$$B_{ij} = 0,$$

$$P_j = P_j$$

Next, for $A_{ij} > 0$ then calculate the remaining cut as follows:

$$c_u = \left(a_{ij}l_i - A_{zj}l_z)\right) \times B_{zj}w_z,$$

$$c_v = \left(a_{ij}l_i\right) \times (k_{ij} - B_{zj}w_z)). \tag{19}$$

while for others, calculate with **Equation (17)**.

Step 7. Calculate $r = n - 1$, if $r > 0$ then proceed to Step 8.

Step 8. Check a_{rj}, if $a_{rj} > 0$ then form a new pattern $(j = j + 1)$ and proceed to Step 9.

Step 9. If $a_{rj} \geq b_{rj}$ then calculate with, For $z = 1, 2, \ldots, r-1$, calculate $a_{zj} = a_{zj}$, and $b_{zj} = b_{zj-1}$.

For $z = r$, calculate $a_{zj} = a_{zj} - 1$ if $a_{zj} > 0$ then $b_{zj} = \lfloor \frac{W}{w_z} \rfloor$.

Otherwise for else $b_{zj} = 0$.

For $z = r+1, \ldots, n$, calculate a_{zj} and b_{zj} using **Equation (12) and (13)**.

However, for $a_{rj} < b_{rj}$ calculate with,

For $z = 1, 2, \ldots, r-1$, calculate $a_{zj} = a_{zj-1}$, and $b_{zj} = b_{zj-1}$.

For $z = r$, calculate $a_{zj} = a_{zj-1}$ and $b_{zj} = b_{zj} - 1$. Then repeat Step 5.

For $z = r+1, \ldots, n$, calculate a_{zj} and b_{zj} using **Equation (12) and (13)**.

Then repeat Step 5.

Step 10. Else, calculate $r = r - 1$ and repeat Step 7.

Step 11. Stop.

3. RESULTS AND DISCUSSION

Writing the results and discussion can be separated into different subs or can also be combined into one sub. The summary of results can be presented in the form of graphs and figures. The results and discussion sections must be free from multiple interpretations. The discussion must answer research problems, support, and defend answers with results, compare with relevant research results, state the study's limitations, and find novelty.

Handal Karya Buana shop is one of the glass cutting shops in Pekanbaru. Handal Karya Buana Store is located on Seokarno-Hatta Street Number 98A, Tobek Godang, Tampan District, Pekanbaru City. This shop was established at the end of January 2021. There are several types of glass starting stock used by Handal Karya Buana, which are distinguished by the size and thickness of each stock. The types of initial stock sizes with the same thickness can be seen in **Table 2**.

Table 2. Types of Initial Stock Sizes Table

No.	Stock size (cm)	Thick (mm)
1	183×122	5
2	240×180	5
3	152×122	5

Handal Karya Buana carries out production based on many incoming requests, so sometimes it is difficult to fulfill requests due to the type of size and number of requests that cannot be predicted. When getting a new order request, it is necessary to plan the right stock cutting technique. Many possible cutting patterns are feasible to cut the initial stock. This cutting pattern must be able to meet consumer demand from the type of size and quantity of demand, but not cause the store to experience losses such as using too much initial stock or too much remaining cutting stock generated.

Moving on from that problem, this article discusses planning for the stock cutting problem using the pattern generation technique which is then solved using the simplex method with the help of the Solver Table software from Microsoft Excel. The following is an example of the demand data received by the Handal Karya Buana store which contains the type of request and the number of units of demand can be seen in **Table 3**.

Table 3. Types of Request and the Number of Units of Demand Table

No.	Unit size (cm)	Demand (mm)
1	56×15	100
2	120×100	20
3	130×117	20

This study presents the formulation of a mathematical model for the optimization of a 2-dimensional stock cutting problem using the pattern generation method and will be solved using Solver Table software. The use of the model in this case gives an overview in planning the stock cutting pattern so as to optimize the existing variables.

Modeling of the stock cutting problem is made based on decision variables, constraint functions, and objective functions. The decision variable in this problem is the type of demand glass. Then the constraint or limitation function in this problem is the number of order requests that must be fulfilled. The objective function in this problem is to minimize the amount of initial stock usage.

a. Decision Variable

The decision variable for this stock cutting problem is formed based on the number of types of glass cut from an initial stock. In this case using a sample of the type of glass demand size received by the Handal Karya Buana store as a decision variable. The variable used is x_j with j depending on the number of patterns generated from various types of stock sizes. The following decision variables are,

x_j ≔ Number of initial stocks cut with the j^{th} pattern

b. Constraint Function

The constraints in this case are as follows:

i. Constraint 1

The number of units produced from the j^{th} cutting pattern with the i^{th} request size is not less than the number of requests for the i^{th} unit size. Constraint 1 is as follows:

$$\sum_{j=1}^{n} p_{ij}x_j \geq d_i, \quad for\ i = 1,2,3$$

ii. Constraint 2

A non-negative constraint that shows all results are non-negative and integer numbers. Constraint 2 is as follows:

$$x_j \geq 0 \quad and\ integer.$$

c. Objective Function

The objective function in this case is to minimize the use of initial stock, which is as follows:

$$z = \sum_{j=1}^{n} x_j.$$

The general form still cannot be solved using the simplex method, because the coefficient value on the constraint function is unknown. The coefficient value of p_{ij} is obtained by using the pattern generation method. The following P is calculated for this case using the pattern generation method algorithm with the Branch and Bound modification discussed in the previous section.

Based on data from Handal Karya Buana, the initial stock size is obtained $L_1 = 183, W_1 = 122, L_2 = 240, W_2 = 180$, and $L_3 = 152, W_3 = 122$. Next sort the size of the demand unit from largest to smallest namely, 130×117, 120×100 and 56×15 along with each demand size. The size of the demand unit obtained by $l_1 > l_2 > l_3$ and $w_1 > w_2 > w_3$ is as follows:

$l_1 = 130\ cm,\ w_1 = 117\ cm,$

$l_2 = 120\ cm,\ w_2 = 100\ cm,$

$l_3 = 56\ cm,\ w_3 = 15\ cm.$

with the number of requests based on the size of the sequential request, namely $d_1 = 20$ sheets, $d_2 = 20$ sheets, and $d_3 = 100$ sheets.

Next, using the pattern generation technique algorithm, calculate (P_1) for stock of Type 1 with **Equation (11)**

$$a_{11} = \left\lfloor \frac{L_1}{l_1} \right\rfloor = \left\lfloor \frac{183}{130} \right\rfloor = 1,$$

$$b_{11} = \left\lfloor \frac{W_1}{w_1} \right\rfloor = \left\lfloor \frac{122}{117} \right\rfloor = 1,$$

$$a_{21} = \left\lfloor \frac{L_1 - (a_{11})(l_1)}{l_2} \right\rfloor = 0,$$

$$b_{21} = 0,$$

$$a_{31} = \left\lfloor \frac{L_1 - \left((a_{11})(l_1) + (a_{21})(l_2)\right)}{l_3} \right\rfloor = 0,$$

$$b_{31} = 0,$$

obtained $P_1 = [1 \quad 0 \quad 0]^T$ then calculate the remaining cutting P_1 against the stock length (c_u),

$$c_u = (L_1 - ((a_{11})(l_1) + \left((a_{21})(l_2) + \left((a_{31})(l_3)\right)\right) \times W_1$$

$$c_u = (183 - \left((1)(130) + (0)(120) + (0)(56)\right) \times 122$$

$$c_u = 53 \times 122 = 6466$$

because $(L - \sum_{i=1}^{n} a_{ij}\, l_i) < w_i$ and $W_i > l_i$ for $i = 3$, so that a new cutting pattern can be formed for using **Equation (15)**,

$$A_{31} = \left\lfloor \frac{L_1 - \sum_{i=1}^{n} a_{ij} l_i}{l_3} \right\rfloor$$

$$A_{31} = \left\lfloor \frac{(183) - ((1)(130))}{15} \right\rfloor$$

$$A_{31} \quad \left\lfloor \frac{53}{15} \right\rfloor = 3,$$

$$B_{31} = \left\lfloor \frac{W_1}{l_3} \right\rfloor$$

$$B_{31} = \left\lfloor \frac{122}{56} \right\rfloor = 2,$$

then, recalculate the remaining cut P_1 using **Equation (19)**.

$$c_u = (L_1 - \sum_{i=1}^{n} a_{ij}\, l_i - A_{31} w_3) \times (B_{31} l_3)$$

$$c_u = (53 - (3)(15)) \times ((2)(56))$$

$$c_u = 8 \times 112 = 896,$$

$$c_v = (L_1 - \sum_{i=1}^{n} a_{ij}\, l_i) \times (W_1 - (B_{31})(l_3))$$

$$c_v = (53) \times (122 - (2)(56))$$

$$c_v = 53 \times 10 = 530.$$

$P_1 = [1\ 0\ 6\ ]^T$ was obtained with a temporary cutting residue of $1426\ \text{cm}^2$. Next, calculate the remaining cutting against the stock width (c_v),

$$c_v = ((a_{11})(l_1)) \times k_{11}$$

$$c_v = ((1)(130)) \times (122 - (1)(117))$$

$$c_v = 130 \times 5 = 650$$

since $((a_{11})(l_1))) > l_z$, while $k_{11} < w_z$ for $z = 2{,}3$, there is no additional new pattern cutting for P_1 against c_v. We get $P_1 = [1\ 0\ 6\ ]^T$ with a total remaining cut of $c_1 = 2076\ \text{cm}^2$. Next, based on step 8, check a_{rj}, with $r = 3 - 1 = 2$, and $j = 1$. Since $a_{21} = 0$ then, $r = 2 - 1 = 1$ and $j = 1$, so that $a_{11} = 1 > 0$ a new pattern can be formed ($j = 2$),

$$\begin{aligned} a_{12} &= a_{11} - 1 = 1 - 1 = 0, \\ b_{12} &= 0, \\ a_{22} &= \left\lfloor \frac{L_1 - (a_{12})(l_1)}{l_2} \right\rfloor = 1, \\ b_{22} &= \left\lfloor \frac{W_1}{w_2} \right\rfloor = \left\lfloor \frac{122}{100} \right\rfloor = 1, \\ a_{32} &= \left\lfloor \frac{L_1 - \left((a_{12})(l_1) + (a_{22})(l_2)\right)}{l_3} \right\rfloor = 1, \\ b_{32} &= \left\lfloor \frac{W_1}{w_3} \right\rfloor = \left\lfloor \frac{122}{15} \right\rfloor = 8, \end{aligned}$$

obtained $P_2 = [0\ 1\ 8\ ]^T$ then calculate the remaining cut P_2 against the stock length (c_u),

$$\begin{aligned} c_u &= (L_1 - ((a_{12})(l_1) + \left((a_{22})(l_2) + \left((a_{32})(l_3)\right)\right) \times W_1 \\ c_u &= (183 - \left((0)(130) + (1)(120) + (1)(56)\right) \times 122 \\ c_u &= 7 \times 122 = 854 \end{aligned}$$

since $(L - \sum_{i=1}^{n} a_{ij}\, l_i) < w_i$ for $i = 1,2,3$, there is no additional cutting of the new pattern so that, $P_2 = [0\ 1\ 8\ ]^T$, with a temporary remaining cutting of 854 cm^2. Next, calculate the remaining cutting against the stock width (c_v),

$$\begin{aligned} c_v &= ((a_{22})(l_2)) \times k_{22} \\ c_v &= ((1)(120)) \times (122 - (1)(100)) \\ c_v &= 120 \times 22 = 2640 \\ c_v &= ((a_{32})(l_3)) \times k_{32} \\ c_v &= ((1)(56)) \times (122 - (8)(15)) \\ c_v &= 56 \times 2 = 112 \end{aligned}$$

since $\left((a_{32})(l_3)\right) < l_z$ and $k_{32} > w_z$ for $z = 1,2$, there is no addition of new pattern cuts. However, for $\left((a_{22})(l_2)\right) > l_z$ and $k_{22} > w_z$ for $z = 3$ a new cutting pattern can be formed for P_2 using **Equation (18)**,

$$\begin{aligned} A_{32} &= \left\lfloor \frac{a_{22} l_2}{l_3} \right\rfloor \\ A_{32} &= \left\lfloor \frac{(1)(120)}{56} \right\rfloor \\ A_{32} &= \left\lfloor \frac{120}{56} \right\rfloor = 2, \\ B_{32} &= \left\lfloor \frac{k_{22}}{w_3} \right\rfloor \\ B_{32} &= \left\lfloor \frac{22}{15} \right\rfloor = 1, \end{aligned}$$

then recalculate the remaining cutting P_2 using **Equation (19)**:

$$\begin{aligned} c_u &= (a_{22} l_2 - A_{32} l_3) \times (B_{32} w_3) \\ c_u &= ((1)(120) - (2)(56)) \times ((1)(15)) \\ c_u &= 8 \times 15 = 120, \\ c_v &= (a_{22} l_2) \times (k_{22} - B_{32} w_3) \end{aligned}$$

$$c_v = ((1)(120)) \times (22 - (1)(15))$$
$$c_v = 120 \times 7 = 840.$$

Obtained $P_2 = [0 \quad 1 \quad 10]^T$ with a total cutting residue of $c_2 = 1926\ cm^2$. Next, based on Step 8, check $a_{rj}$with $r = 3 - 1 = 2$ dan $j = 2$. Since $a_{22} > 0$ a new pattern can be formed ($j = 3$),

$$a_{13} = a_{21} = 0$$
$$b_{13} = 0,$$
$$a_{23} = a_{22} - 1 = 1 - 1 = 0,$$
$$b_{23} = 0,$$
$$a_{33} = \left\lfloor \frac{L_1 - ((a_{13})(l_1) + (a_{23})(l_2))}{l_3} \right\rfloor = \left\lfloor \frac{183 - ((0)(130) + (0)(120))}{56} \right\rfloor = 3,$$
$$b_{33} = \left\lfloor \frac{W_1}{w_3} \right\rfloor = \lfloor \frac{122}{15} \rfloor = 8,$$

obtained $P_3 = [0 \quad 0 \quad 24]^T$, then calculate the remaining cut P_3 against the stock length (c_u),

$$c_u = (L_1 - ((a_{13})(l_1) + \left((a_{23})(l_2) + ((a_{33})(l_3))\right) \times W_1$$
$$c_u = (183 - \left((0)(130) + (0)(120) + (3)(56)\right) \times 122$$
$$c_u = 15 \times 122 = 1830$$

since $(L_1 - \sum_{i=1}^{n} a_{ij} l_i) < w_i$ and $W_1 > l_i$ for $i = 3$, a new cutting pattern for P_3 can be formed using **Equation (15)**,

$$A_{33} = \lfloor \frac{L_1 - \sum_{i=1}^{n} a_{ij} l_i}{l_3} \rfloor$$
$$A_{33} = \lfloor \frac{(183) - ((3)(56))}{15} \rfloor$$
$$A_{33} = \lfloor \frac{15}{15} \rfloor = 1,$$
$$B_{33} = \lfloor \frac{W_1}{l_3} \rfloor$$
$$B_{33} = \lfloor \frac{122}{56} \rfloor = 2,$$

then recalculate the remaining cutting P_3 using **Equation (19)**,

$$c_u = \left(L_1 - \sum_{i=1}^{n} a_{ij}\, l_i - A_{33} w_3\right) \times (B_{33} l_3)$$
$$c_u = (15 - (1)(15)) \times ((2)(56))$$
$$c_u = 0 \times 112 = 0,$$
$$c_v = (L_1 - \sum_{i=1}^{n} a_{ij} l_i) \times (W_1 - (B_{33})(l_3))$$
$$c_v = (15) \times (122 - (2)(56))$$
$$c_v = 15 \times 10 = 150.$$

$P_3 = [0\ 0\ 26\ ]^T$ is obtained with a temporary cutting residue of $150\ \mathrm{cm}^2$. It is obtained with a temporary cutting residue of (c_v),

$$c_v = ((a_{33})(l_3)) \times k_{33}$$
$$c_v = ((3)(56)) \times (122 - (8)(15))$$
$$c_v = 168 \times 2 = 336$$

since $((a_{33})(l_3))) > l_z$, while $k_{33} < w_z$ for $z = 2$, there is no additional new pattern cutting for P_3 with respect to c_v. We get $P_3 = [0\ 0\ 26\ ]^T$ with a total remaining cut of $c_3 = 486\ \mathrm{cm}^2$. Next based on step 8, check a_{rj} with $r = 3 - 1 = 2$ and $j = 3$. Since $a_{23} = 0$, then proceed based on Step 10, $r = 2 - 1 = 1$, and

$j = 3$, so that $a_{13} = 0$, and no new pattern can be generated. Next calculate using stock Type 2, with the initial pattern ($j = 4$), do the pattern generator calculation as the previous step. With the pattern generator algorithm, $j = 10$ or as many as 10 glass stock cutting patterns for the case at Handal Karya Buana store. Next, substitute the values of $P_1, P_2, \dots, P_{10}$ into **Equation (10)** so that P is obtained as follows:

$$P = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0\,0 \quad 1 \quad 0 \quad 0 \quad 2 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0\; 6 \quad 10 \quad 26 \quad 29 \quad 20 \quad 34 \quad 51 \quad 2 \quad 6 \quad 20\,]. \qquad (20)$$

Based on **Equation (20)**, 10 cutting patterns are obtained, so the objective function can be written as:

$$min \quad z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}.$$

Then, by substituting the cutting pattern p_{ij} the case constraint function, by substituting the cutting pattern p_{ij} so that the constraint function can be written into:

$$\begin{aligned} x_1 + x_4 + x_8 &\geq 20 \\ x_2 + 2x_5 + x_6 + x_9 &\geq 20 \\ 6x_1 + 8x_2 + 24x_3 + 29x_4 + 20x_5 + 34x_6 + 51x_7 + 2x_8 + 6x_9 + 20x_{10} &\geq 100 \\ x_i \geq 0, \qquad \text{for} \quad i &= 1, 2, \dots, 10 \end{aligned}$$

The solution of the mathematical model of the 2-dimensional stock cutting problem using the pattern generator method was carried out with the help of the Solver Table software on Microsoft Excel because of the large number of decision variables and constraints. The results of this study are in the form of recommendations and suggestions for stores to determine better stock cutting patterns to fulfill orders received by comparing conventional methods used by stores. The goal to be optimized in this problem is to minimize the use of stock that can save store capital. The solution of this case using Microsoft Excel is presented in **Table 4**.

Table 4. The Solution Table

Pattern	Unit 1	Unit 2	Unit 3	Remaining Cutting	Stock Requirement
1	1	0	6	2076	0
2	0	1	10	1926	0
3	0	0	26	486	0
4	1	0	29	3630	0
5	0	2	20	2400	10
6	0	1	34	2640	0
7	0	0	51	360	0
8	1	0	2	1654	20
9	0	1	6	1504	0
10	0	0	20	1744	0
Total Units	20	20	240		
Total Remaining Cutting				57080	
Total Stock Requirement					30

Based on **Table 4**, if the store wants to save production costs using initial stock, the store can use as much as 10 initial stock measuring 240 cm $\times$ 180 cm cm cut with Pattern 5 and as much as 20 initial stock measuring 152 cm $\times$ 122 cm cut with Pattern 8. The resulting cutting residue is 57080 cm^2.

4. CONCLUSIONS

The pattern generation method produces a matrix containing feasible stock cutting patterns. The cutting pattern is then converted into a linear equation which is then solved using the simplex method. Solving the cutting stock problem using the pattern generation method can provide better cutting pattern planning. In planning the right stock cutting pattern, there are many possible cutting patterns. So that by using the pattern generation method, many of the resulting cutting patterns are increasingly narrowed or less than the patterns generated at the beginning, making it easier to solve the case mathematically.

ACKNOWLEDGMENT

Thank you to Handal Karya Buana for helping us during the article writing process, providing input, explanations and data for this article.

REFERENCES

[1] P. C. Gilmore and R. E. Gomory, *A linear programming approach to the cutting-stock problem*, vol. 9. 1961.
[2] R. W. Haessler and P. E. Sweeney, “Cutting stock problems and solution procedures,” *Eur. J. Oper. Res.*, vol. 54, no. 2, pp. 141–150, 1991, doi: 10.1016/0377-2217(91)90293-5.
[3] S. Octarina, M. Radiana, and P. B. J. Bangun, “Implementation of pattern generation algorithm in forming Gilmore and Gomory model for two dimensional cutting stock problem,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 300, no. 1, 2018, doi: 10.1088/1757-899X/300/1/012021.
[4] S. Boualem and M. Abbas, “Contribution to Solving A Two-Dimensional Cutting Stock Problem with Two Objectives,” *Econ. Comput. Econ. Cybern. Stud. Res.*, vol. 56, no. 2, pp. 179–194, 2022, doi: 10.24818/18423264/56.2.22.12.
[5] B. I. Erdem and R. Kasimbeyli, “a Two-Step Math Heuristic Solution Approach for the Two-Dimensional Cutting Stock Problem,” *J. Nonlinear Convex Anal.*, vol. 24, no. 4, pp. 681–699, 2023.
[6] P. B. J. Bangun, S. Octarina, and A. P. Pertama, “Implementation of branch and cut method on n-sheet model in solving two dimensional cutting stock problem,” *J. Phys. Conf. Ser.*, vol. 1282, no. 1, 2019, doi: 10.1088/1742-6596/1282/1/012012.
[7] M. Dodge, S. A. MirHassani, and F. Hooshmand, “Solving two-dimensional cutting stock problem via a DNA computing algorithm,” *Nat. Comput.*, vol. 20, no. 1, pp. 145–159, 2021, doi: 10.1007/s11047-020-09786-3.
[8] M. Atika, B. P. Silalahi, and F. Bukhari, “Pattern Generation for Three Dimensional Cutting Stock Problem,” *JTAM (Jurnal Teor. dan Apl. Mat.*, vol. 6, no. 4, p. 991, 2022, doi: 10.31764/jtam.v6i4.9933.
[9] P. Caricato and A. Grieco, “An Application of Industry 4.0 to the Production of Packaging Films,” *Procedia Manuf.*, vol. 11, no. June, pp. 949–956, 2017, doi: 10.1016/j.promfg.2017.07.199.
[10] A. Rahman, I. Hasbiyati, and M. D. H. Gamal, “Approximation Method for Solving TwoDimensional Cutting-Stock Problems Abstract :,” vol. 19, no. 1, pp. 29–34, 2023, doi: 10.9790/5728-1901022934.
[11] N. P. Rodrigo, W. B. Daundasekera, and A. I. Perera, “One-Dimensional Bin-Packing Problems with Branch and Bound Algorithm,” *Int. J. Discret. Math.*, vol. 3, no. 2, pp. 36–40, 2018, doi: 10.11648/j.dmath.20180302.12.
[12] S. Lomate, B. Rajiv, P. D. Pantawane, and B. B. Ahuja, “Greedy Algorithm to generate cutting patterns for Cutting Stock Problem (1D and 2D),” *Ind. Eng. J.*, vol. 13, no. 4, 2020, doi: 10.26488/iej.13.4.1223.
[13] S. Altin, T. Aydilek, U. Sirvan, D. Kesikburun, A. Oner, and N. Kutup, “Three Dimensional Cutting Stock Problem in Mattress Production: A Case Study,” *Proc. Int. Symp. Prod. Res. 2018*, pp. 949–960, 2019, doi: 10.1007/978-3-319-92267-6.
[14] F. S. Hillier, G. J. Lieberman, and F. S. Hillier, *Introduction to Operation Research.*, 9th ed. New Jersey: McGraw-Hill Education, 2010.
[15] W. L. Winston, *Operations Research: Applications and Algorithms*, 4th ed. Belmont: Thomson Brooks/ Cole, 2004.
[16] M. D. H. Gamal, “Program Linear dan Integer,” 2024. https://repository.unri.ac.id/handle/123456789/11313
[17] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, “Linear Programming and Network Flows,” *J. Oper. Res. Soc.*, vol. 29, no. 5, p. 510, 1978, doi: 10.2307/3009778.
[18] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*. New Jersey: John Wiley & Sons, 2013.