

## Implementasi Untuk Optimasi Biaya Produksi Dan Target Penjualan ( Toko Roti Tiga Bintang ) Menggunakan Metode Goal Programming

Mahtahul Hakimah<sup>1</sup>, Danang Haryo Sulaksono<sup>2</sup>, Hanafi Sasmita<sup>3</sup>  
<sup>1,2</sup>Jurursan Tehnik Informatika, Fakultas Elektro dan Teknologi Informasi  
Institut Teknologi Adhi Tama Surabaya

Email: [hakimah.mafta@itats.ac.id](mailto:hakimah.mafta@itats.ac.id), [danang\\_h\\_s@itats.ac.id](mailto:danang_h_s@itats.ac.id), [nafisony4@gmail.com](mailto:nafisony4@gmail.com)

---

### ABSTRACT

The three-star bakery is a home industry that produces rolls and donuts every day. So far, the problems faced by this bakery are profits that have not been maximized and income from donut sales is very minimal, using the Goal Programming method with a case study dataset at the Tiga Bintang Bakery, for reference attributes that are the reference in this study are raw material costs, direct, employee salaries and overhead costs, the main component of a linear programming problem is an objective function in the form of a maximum or minimum criterion. of the model. Therefore, the objective function of each target or goal is part of the objective function of the overall goal programming problem. The data used in this optimization is data from the Tiga Bintang Bakery which is produced per day from indirect raw material costs, employee salaries, overhead costs and profits for each roll and donut per day, Determining the weight value taken from each goal which has a level the priority of the same importance to be achieved, so there is no priority which target must be achieved first, the results of the optimization between rolls and donuts after testing the system get results from  $x_1$  2,273 and  $x_2$  681 which are calculated to get a total profit of 1,881,911 the previous one was 1,834,000

---

### Article History

Received 2020-10-02  
Revised 2020-10-16  
Accepted 2020-10-23

### Key words

*Linear Programming, Goal Programming*

### ABSTRAK

Toko roti tiga bintang merupakan industri rumah tangga yang memproduksi Roti gulung dan donat setiap hari. Selama ini permasalahan yang dihadapi oleh toko roti ini adalah keuntungan yang belum maksimal dan pendapatan penjualan donat sangat minim, dengan menggunakan metode Goal Programming dengan dataset studi kasus pada Toko Roti Tiga Bintang, untuk atribut acuan yang menjadi acuan dalam penelitian ini adalah biaya bahan baku tidak langsung, gaji karyawan dan biaya overhead, Komponen utama dari masalah program linier (linier programming) adalah fungsi tujuan yang berupa kriteria maksimum atau minimum Tujuan dari goal programming adalah meminimumkan deviasi atau penyimpangan dari semua target yang diinginkan, dengan mempertimbangkan semua kendala dari target serta kendala dari modelnya. Oleh karena itu, fungsi tujuan setiap target atau goal merupakan bagian dari fungsi tujuan masalah goal programming secara keseluruhan. Data yang digunakan pada optimasi ini adalah data dari Toko Roti Tiga Bintang yang di produksi perhari dari biaya bahan baku tidak langsung, gaji karyawan, biaya overhead dan keuntungan tiap roti gulung dan donat perhari, Menentukan nilai bobot diambil dari masing-masing goal yang mempunyai tingkat prioritas kepentingan yang sama untuk dicapai, jadi tidak ada prioritas mana target yang harus dicapai lebih dulu, Hasil dari pengoptimalan antara roti gulung dan donat sesudah di lakukan pengujian sistem di dapatkan hasil dari  $x_1$  2.273 dan  $x_2$  681 yang di hitung mendapatkan total keuntungan 1.881.911 yang sebelumnya 1.834.000

---

### PENDAHULUAN

Toko roti tiga bintang merupakan industri rumah tangga yang memproduksi Roti gulung dan donat setiap hari. Selama ini permasalahan yang dihadapi oleh toko roti ini adalah keuntungan yang belum maksimal dan pendapatan penjualan donat sangat minim karena setiap harinya hanya terjual rata-rata 30 biji Kegiatan pembuatan roti pada Toko Roti Tiga Bintang menginginkan agar meningkatkan keuntungan dan penjualan, karena pada pembuatan donat yang penjualannya setiap hari tidak bisa maksimal,

Ada juga faktor-faktor yang mempengaruhi seperti biaya tidak langsung, gaji karyawan dan biaya overhead, Pemilik toko roti menginginkan solusi yang tepat untuk bisa mengatasi masalah pada usaha yang sedang dilakukan saat ini, Pada dasarnya tujuan dari setiap perusahaan baik bergerak dibidang manufaktur maupun jasa adalah memperoleh keuntungan yang maksimal atau menekan

biaya yang paling minimal. Produksi banyak tidak selalu menjamin memperoleh keuntungan yang maksimal. Pada saat ini banyak bermunculan perusahaan yang bergerak dibidang industri makanan, dalam perkembangannya perusahaan-perusahaan mengalami persaingan yang sangat ketat diantara perusahaan yang memproduksi sejenis. Perubahan-perubahan yang begitu cepat dalam bisnis yang menuntut perusahaan harus lebih mampu beradaptasi, mempunyai ketahanan, mampu melakukan perubahan arah dengan cepat dan memusatkan perhatian pada konsumen. Dalam suasana bisnis seperti sekarang ini perusahaan harus mampu menjadi mitra kerja yang handal bagi para konsumen ditengah persaingan yang semakin ketat. Industri makanan yang dalam persaingannya yaitu mengenai produk-produk roti dan kue saat ini.

Meninjau permasalahan tersebut usulan penelitian ini adalah bertujuan untuk memaksimalkan keuntungan dan meningkatkan pendapatan penjualan donat, maka dari itu membutuhkan metode optimasi yang tepat untuk mengatasi permasalahan tersebut Tanpa adanya kegiatan optimasi untuk keuntungan dan memaksimalkan pendapatan penjualan donat maka pemilik toko tidak akan bisa memaksimalkan penjualan tiap produk-produk tersebut. Untuk itu diperlukan metode optimasi yang tepat karena optimasi dilakukan untuk 2 target yaitu roti gulung dan donat maka pada usulan penelitian ini menggunakan metode optimasi yang dipilih adalah Goal Programming.

Penelitian sebelumnya yang digunakan pada masalah ini adalah penilaian berkelanjutan dan peningkatan efisiensi keberlanjutan dalam Industri Kimia yang dilakukan oleh Luiz Carlos Barbosaa dan Luiz Flavio Autran Monteiro Gomes dengan menggunakan metode Goal Programming dengan dataset studi kasus pada Toko Roti Tiga Bintang, untuk atribut acuan yang menjadi acuan dalam penelitian ini adalah biaya bahan baku tidak langsung, gaji karyawan dan biaya *overhead*

## TINJAUAN PUSTAKA

### Pemrograman Linier

Pemrograman linear (linear programming) adalah teknik pengambilan keputusan untuk memecahkan masalah mengalokasikan sumber daya yang terbatas diantara berbagai kepentingan seoptimal mungkin. Pemrograman linear merupakan salah satu metode dalam riset operasi yang memungkinkan para manajer mengambil keputusan dengan menggunakan pendekatan analisis kuantitatif. Teknik ini telah diterapkan secara luas pada berbagai persoalan dalam perusahaan, untuk menyelesaikan masalah yang berkaitan dengan penugasan karyawan, penggunaan mesin, distribusi, dan pengangkutan, penentuan kapasitas produk, ataupun dalam penentuan portofolio investasi

Bentuk umum dari Pemrograman Linier :

$$\text{maks / min } z(x_1, x_2, \dots, x_n) = \sum_{j=1}^n c_j x_j \dots \dots \dots (2.1)$$

Dengan kendala,

$$\sum_{j=1}^n a_{ij} x_j \begin{pmatrix} \leq \\ = \\ \geq \end{pmatrix} b_i, i = 1, 2, 3, \dots, m \dots \dots \dots (2.2)$$

Dan

$$x_j \geq 0, j = 1, 2, 3, \dots, n \dots \dots \dots (2.3)$$

Keterangan:

$z$  = fungsi tujuan.

$x_j$  = jenis kegiatan (variabel keputusan).

$a_{ij}$  = kebutuhan sumber daya  $i$  untuk menghasilkan setiap unit kegiatan  $j$ .

$b_i$  = jumlah sumber daya  $i$  yang tersedia.

$c_j$  = kenaikan nilai  $Z$  jika ada pertambahan satu unit kegiatan  $j$ .

$a, b, c$  = disebut juga sebagai parameter model.

$m$  = jumlah sumberdaya yang tersedia.

$n$  = jumlah kegiatan.

### Simpleks Dasar

Metode Simpleks merupakan suatu cara penyelesaian masalah program linier yang diperkenalkan oleh Dantzig pada tahun 1947, yakni suatu cara penyelesaian dengan menggunakan metode “*Row Operation matrix*” yang khusus disebut “*Pivot Operation*”. Metode ini telah terbukti efisien untuk memecahkan persoalan program linier dalam skala besar. Kecuali untuk persoalan yang kecil, metode ini selalu dikerjakan dengan menggunakan komputer

### BIG-M

Metode Big M digunakan untuk menyelesaikan fungsi-fungsi dalam program linier yang tidak berada dalam bentuk baku atau standar (bentuk standar adalah memaksimalkan Z sesuai dengan kendala fungsional dalam bentuk  $\leq$  dan kendala nonnegativitas di semua variabel) dan salah satu contoh masalah dalam kendala fungsional adalah bila fungsi dalam bentuk-bentuk  $=$  atau  $\geq$  atau bahkan ruas kanan yang negatif.

Teknik pendekatan khusus tersebut dengan cara menambahkan variabel dummy (variabel artifisial) pada kendala fungsional dan teknik ini disebut dengan teknik variabel artifisial. Ada pun prosedur mendapatkan BF awal pada kendala fungsional adalah :

- a. Gunakan teknik variabel artifisial

Tambahkan variabel artifisial nonnegatif pada fungsi kendala yang belum baku, dan anggaplah variabel artifisial tersebut sebagai salah satu variabel

- b. Tugaskan pinalty yang besar

Berilah nilai variabel artifisial dengan nilai  $> 0$  sehingga koefisien variabel artifisial menjadi M (big m) secara simbolik yang menunjukkan bahwa variabel artifisial tersebut memiliki angka positif raksasa (dan pengubahan atas variabel artifisial bernilai 0 (variabel nonbasis) dalam solusi optimal disebut metode big m)

contoh di bawah ini. Bentuk Umum :

$$\begin{aligned} \text{Min. } z &= 4x_1 + x_2 \text{ Terhadap: } 3x_1 + x_2 \\ &= 3 \\ 4x_1 + 3x_2 &\geq 6 \quad x_1 + 2x_2 \leq 4 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Kendala 1 dan 2 tidak mempunyai slack variables, sehingga tidak ada variabel basis awal. Untuk berfungsi sebagai variabel basis awal, pada kendala 1 dan 2 ditambahkan masing-masing satu variabel buatan (artificial variable). Maka bentuk baku Big M-nya adalah:

$$\text{Min. } z = 4x_1 + x_2 + MA_1 + MA_2$$

$$\text{Terhadap : } 3x_1 + x_2 + A_1 = 3$$

$$4x_1 + 3x_2 - s_1 + A_2 = 6 \quad x_1 + 2x_2 + s_2 = 4$$

$$x_1, x_2, s_1, s_2 \geq 0$$

### Goal Programming

Komponen utama dari masalah program linier (linier programming) adalah fungsi tujuan yang berupa kriteria maksimum atau minimum. Tujuan dari goal programming adalah meminimumkan deviasi atau penyimpangan dari semua target yang diinginkan, dengan mempertimbangkan semua kendala dari target serta kendala dari modelnya. Oleh karena itu, fungsi tujuan setiap target atau *goal merupakan* bagian dari fungsi tujuan masalah goal programming secara keseluruhan.

Pasangan variable deviasi  $s_1^-$  dan  $s_1^+$  akan dimunculkan pada ruas kiri persamaan kendala yang dirumuskan dari fungsi target. Jadi misalkan fungsi targetnya adalah :

$f_1(x_j) \leq \geq = f_1$  , maka persamaan kendala yang berkaitan dengan target tersebut dirumuskan sebagai :  $f_1(x_j) + s_1^- - s_1^+ = f_1$

Dengan:  $s_1^-$  = besarnya deviasi atau penyimpangan di bawah target.

$s_1^+$  = besarnya deviasi atau penyimpangan di atas target.

Apabila ada *goal* yang menyatakan bahwa batas atas target sama sekali tidak boleh dilanggar, maka fungsi kendala targetnya menjadi :

$f_1(x_j) + s_1^- = f_1$  (sama sekali tidak diijinkan lebih dari  $f_1$ )

Apabila ada *goal* yang menyatakan bahwa batas bawah target sama sekali tidak boleh dilanggar, maka fungsi kendala targetnya menjadi :

$f_1(x_j) - s_1^+ = f_1$  (sama sekali tidak diijinkan kurang dari  $f_1$ )

Jenis *goal* yang pertama (biasa pernyataan targetnya dinyatakan dengan kata-kata: paling sedikit, setidaknya, tidak kurang dari, minimal dan seterusnya) atau jika fungsi targetnya menggunakan tanda  $\geq$ , maka fungsi tujuannya adalah meminimumkan kekurangan dari target atau variabel *slack*  $s_1^-$  yang merupakan variabel penyimpangan di bawah target.

Jenis *goal* yang kedua (biasanya pernyataan targetnya dinyatakan dengan kata-kata: paling banyak, tidak lebih dari, dan seterusnya) atau jika fungsi targetnya menggunakan tanda  $\leq$ , maka fungsi tujuannya adalah meminimumkan kelebihan dari target atau variabel *surplus*  $s_1^+$  yang merupakan variabel penyimpangan di atas target.

Jenis *goal* yang ketiga (biasanya pernyataan targetnya dinyatakan dengan kata-kata: kalau bias tidak lebih dan tidak kurang), atau jika fungsi targetnya menggunakan tanda  $=$ , maka fungsi tujuannya adalah meminimumkan jumlah kelebihan dan kekurangan dari target atau jumlah variabel penyimpangan  $s_1^- + s_1^+$ .

Berdasarkan tingkat kepentingan dari masing-masing target atau sasaran (*goal*),

*Goal programming* di bedakan dalam 2 kategori, yaitu:

1. **Nonpreemptive goal programming**, jika masing-masing *goal* mempunyai tingkat prioritas kepentingan yang sama untuk dicapai, jadi tidak ada prioritas mana target yang harus dicapai lebih dulu.
2. **Preemptive goal programming**, jika terdapat urutan tingkat prioritas  
 Seperti telah disebutkan di atas, *goal programming* merupakan perluasan dari model program linier, sehingga seluruh asumsi, notasi, formulasi model matematis, prosedur perumusan model dan metode penyelesaiannya tidak berbeda dengan model program linier.

#### FORMULASI GOAL PROGRAMMING

Minimasi  $Z = \sum_{i=1}^m P_i (s_1^- + s_1^+)$

Dengan kendala:

$$\begin{aligned} f_1(x_1) - s_1^- - s_1^+ &= f_1 & ; i = 1, 2, \dots, m \\ g_k(x_1) &\geq \leq = b_k & ; k = 1, 2, \dots, k \\ x_1 &\geq 0 & ; j = 1, 2, \dots, n \end{aligned}$$

Keterangan:

- $x_1$  = variabel keputusan  
 $n$  = banyaknya variabel keputusan  
 $f_1(x_1)$  = fungsi target atau *goal*  
 $f_1$  = batas atas atau batas bawah target atau goal  
 $s_1^-$  = variabel deviasi yang menunjukkan penyimpangan di bawah target (variabel *slack*)  
 $s_1^+$  = variabel deviasi yang menunjukkan penyimpangan di atas target (variabel *surplus*)  
 $m$  = banyaknya target atau *goal*

$g_k(x_1)$  = fungsi kendala dalam model

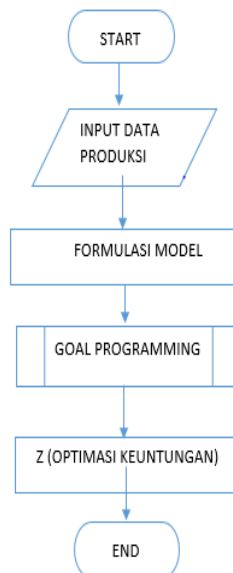
## METODE

### Gambaran Umum

Pada penelitian kali ini di terapkan goal programming untuk di implementasikan pada Toko Roti Tiga Bintang, Salah satu faktor untuk menghasilkan produk yang efektif dan efisien adalah dengan membuat perencanaan produksi yang baik, untuk itu Toko Roti Tiga Bintang membutuhkan perencanaan produksi yang matang guna mengoptimalkan sumber daya yang terbatas untuk mencapai setiap tujuan yang dikehendaki took roti. Dalam hal ini perencanaan produksi adalah 2 produk yaitu roti gulung dan donat sehingggs variabel keputusan yang digunakan adalah :

1.  $X_1$  jumlah produksi roti gulung.
2.  $X_2$  jumlah produksi donat.

Optimasi  $X_1$  dan  $X_2$  digunakan memaksimalkan keuntungan dan meningkatkan pendapatan penjualan donat. keuntungan disini di asumsikan hanya dipengaruhi adalah biaya bahan baku langsung, gaji karyawan dan *overhead*.

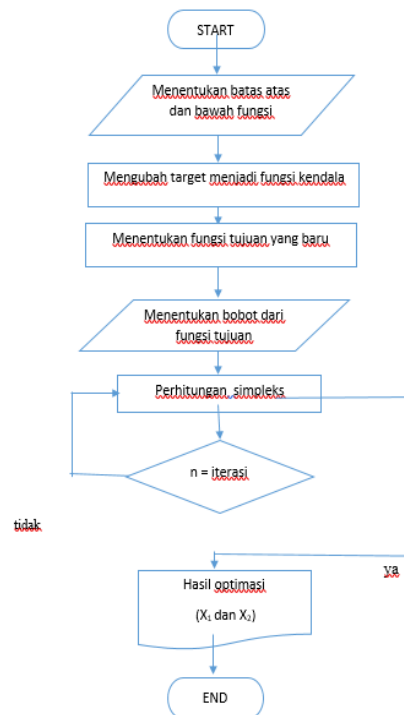


**Gambar 3.1** Flowchart keseluruhan

Gambar 3.1 merupakan flowchart yang digunakan pada optimasi Toko Roti Tiga Bintang. Pertama masukkan kedua variabel produksi roti gulung dan donat yang digunakan untuk mencari optimasi biaya bahan baku dan pendapatan penjualan donat. Kemudian kedua variabel tersebut di optimasi menggunakan Goal Programming. Setelah kedua variabel tersebut dihitung menggunakan Goal programming, akan menunjukkan nilai optimasi dari kedua variabel tersebut

Gambar 3.2 merupakan proses langkah-langkah optimasi menggunakan metode Goal Programming. Proses awal mengumpulkan batas atas dan batas bawah dengan menggunakan variabel  $S^-$  dan  $S^+$ , selanjutnya adalah mengubah target menjadi kendala  $X_1$  dan  $X_2$ , setelah itu adalah menentukan fungsi baru karena target yang telah di ubah menjadi kendala, selanjutnya adalah menentukan bobot dari masing-masing kendala yang sudah ada, yang

terakhir adalah perhitungan menggunakan simpleks yang akan dilanjutkan pembentukan table iterasi 1, 2, 3 dan seterusnya sampai memperoleh hasil yang optimal dari  $X_1$  dan  $X_2$ .



**Gambar 3.2** Flowchart Goal Programming

### Persiapan data

Data yang digunakan pada optimasi ini adalah data dari Toko Roti Tiga Bintang yang di produksi perhari dari biaya bahan baku tidak langsung ,gaji karyawan, biaya *overhead* dan keuntungan tiap roti gulung dan donat perhari akan dijelaskan pada table berikut ini :

Nama Produk	Biaya Bahan Baku tidak langsung (Rp)	Gaji Karyawan (Rp)	Biaya Overhead (Rp)	Total Biaya Produksi (Rp)	Total Keuntungan (Rp)
Roti Gulung (X <sub>1</sub> )	2.200	80	13.000	15.280	545
Donat (X <sub>2</sub> )	367	320	30.000	30.687	943
				45.967	1.488

**Tabel 3.1** Data Perhari Toko Roti Tiga Bintang.

### Menentukan Bobot Dari Fungsi Tujuan

Menentukan nilai bobot diambil dari masing-masing *goal* yang mempunyai tingkat prioritas kepentingan yang sama untuk dicapai, jadi tidak ada prioritas mana target yang harus dicapai lebih dulu

### Formulasi Model

Berikut ini formulasi model dari data yang diatas agar dapat di optimalkan.

Formulasi Model :

Maksimasi :

Goal 1 :  $Z_1 = 545 X_1 + 943 X_2$  ( keuntungan penjualan )

Goal 2 :  $Z_2 = X_2 \geq 50$  ( penjualan donat )

Kendala :

$2.200 X_1 + 367 X_2 \leq 6.800.000$  ( biaya produksi )

$80 X_1 + 320 X_2 = 400.000$  ( gaji karyawan )

$13X_1 + 30X_2 \leq 50.000$  ( biaya overhead )

$X_1, X_2 \geq 0$

Target 1 : Keuntungan  $\geq 2.200.000$

Target 2 : Pendapatan Penjualan Donat  $\geq 50$

### Penyelesaian Model Formulasi Menggunakan Goal Programming

Dari formulasi model diatas maka harus di cari formulasi baru untuk perhitungan Goal Programming yang mengacu pada Target 1 dan Target 2 maka model Goal Programming untuk studi kasus ini adalah :

Minimasi :  $Z = 4 S_1^- + 0 S_1^+ + 3 S_2^- + 0 S_2^+ + 0 S_3 + MR_1 + 0 S_4$  ( tujuan yang sudah diberi nilai bobot )

Kendala :

$545 X_1 + 943 X_2 + S_1^- - S_1^+ = 2.200.000$  ( keuntungan )

$X_2 + S_2^- - S_2^+ = 50$  ( penjualan donat )

$2.200X_1 + 367 X_2 + S_3 = 6.800.000$  ( biaya produksi )

$80 X_1 + 320 X_2 + MR_1 = 400.000$  ( gaji karyawan )

$13X_1 + 30X_2 + S_4 = 50.000$  ( overhead )

$X_1, X_2, S_1^-, S_1^+, S_2^-, S_2^+, R_1, S_3, S_4 \geq 0$

Penyelesaian model ini dapat diselesaikan menggunakan metode simpleks dan Big-M sebagai

beikut :

**Tabel 3.2** Tabel Simpleks Awal

$c_j$		0	0	4	0	3	0	0	m	0		
$c_j$	$x_j/x_j$	$x_1$	$x_2$	$s_1^-$	$s_1^+$	$s_2^-$	$s_2^+$	$s_3$	$r_1$	$s_4$	rhs	rasio
4	$s_1^-$	545	943	1	-1	0	0	0	0	0	2.200.000	2.120
3	$s_2^-$	0	1	0	0	1	-1	0	0	0	50	50
0	$s_3$	2.200	367	0	0	0	0	1	0	0	6.800.000	18.528,6104
m	$r_1$	80	320	0	0	0	0	0	1	0	400.000	1.250
0	$s_4$	13	30	0	0	0	0	0	0	1	50.000	1.666,666



	cj	0	0	4	0	3	0	0	1000	0		
cj	xj/xj	x1	x2	s1-	s1+	s2-	s2+	s3	r1	s4	rhs	rasio
4	s1-	545	943	1	-1	0	0	0	0	0	2200000	2332.97985
3	s2-	0	1	0	0	1	-1	0	0	0	50	50
0	s3	2200	367	0	0	0	0	1	0	0	6800000	18528.6104
1000	r1	80	320	0	0	0	0	0	1	0	400000	1250
0	s4	13	30	0	0	0	0	0	0	1	50000	1666.66667
	zj	82180	323775	4	-4	3	-3	0	1000	0		
	zj-cj	82180	323775	0	-4	0	-3	0	0	0		
	cj	0	0	4	0	3	0	0	1000	0		
cj	xj/xj	x1	x2	s1-	s1+	s2-	s2+	s3	r1	s4	rhs	rasio
4	s1-	545	0	1	-1	-943	943	0	0	0	2152850	2282.97985
0	x2	0	1	0	0	1	-1	0	0	0	50	
0	s3	2200	0	0	0	-367	367	1	0	0	6781650	18478.6104
1000	r1	80	0	0	0	-320	320	0	1	0	384000	1200
0	s4	13	0	0	0	-30	30	0	0	1	48500	1616.66667
	zj	82180	0	4	-4	-323772	323772	0	1000	0		
	zj-cj	82180	0	0	-4	-323775	323772	0	0	0		
	cj	0	0	4	0	3	0	0	1000	0		
cj	xj/xj	x1	x2	s1-	s1+	s2-	s2+	s3	r1	s4	rhs	rasio
4	s1-	309.25	0	1	-1	0	0	0	-2.94688	0	1021250	3302.34438
0	x2	0.25	1	0	0	0	0	0	0.003125	0	1250	5000
0	s3	2108.25	0	0	0	0	0	1	-1.14688	0	6341250	3007.8264
0	s2+	0.25	0	0	0	-1	1	0	0.003125	0	1200	4800
0	s4	5.5	0	0	0	0	0	0	-0.09375	1	12500	2272.72727
	zj	1237	0	4	-4	0	0	0	-11.7875	0		
	zj-cj	1237	0	0	-4	-3	0	0	-1011.79	0		
	cj	0	0	4	0	3	0	0	1000	0		
cj	xj/xj	x1	x2	s1-	s1+	s2-	s2+	s3	r1	s4	rhs	rasio
4	s1-	0	0	1	-1	0	0	0	2.324432	-56.2273	318409.091	
0	x2	0	1	0	0	0	0	0	0.007386	-0.04545	681.818182	
0	s3	0	0	0	0	0	0	1	34.7892	-383.318	1549772.73	
0	s2+	0	0	0	0	-1	1	0	0.007386	-0.04545	631.818182	
0	x1	1	0	0	0	0	0	0	-0.01705	0.181818	2272.72727	
	zj	0	0	4	-4	0	0	0	9.297727	-224.909	1273636.36	
	zj-cj	0	0	0	-4	-3	0	0	-990.702	-224.909		

**Gambar 3.3** Gambar Penyelesaian Untuk Mendapatkan Nilai Optimal

Gambar 3.3 di atas merupakan proses iterasi untuk mendapatkan nilai optimal menggunakan simpleks. Hasil optimal yang diperoleh adalah  $x_1 = 2272.727273 \approx 2273$  atinya roti gulung yang diproduksi sebanyak 2273 buah perhari. Sedangkan  $x_2 = 681,818182 \approx 682$ . Artinya donat yang diproduksi sebanyak 682 buah perhari

### Skenario Pengujian

skenario pengujian ini adalah membandingkan keuntungan sebelum dan sesudah melakukan proses optimasi menggunakan Goal Programming, perbandingan bisa dilihat pada tabel berikut ini :



**Tabel 3.4** perbandingan keuntungan antara sebelum dan sesudah optimasi menggunakan Goal Programming.

$X_i$	Sebelum Optimal	Sesudah Optimal
Roti Gulung	2.500	2.273
Donat	500	682
Keuntungan	1.834.000	1.881.911

## Pengujian Sistem

Pengujian sistem pada formulasi model menggunakan goal programming dan akan ada alur proses perhitungan dari awal hingga akhir dengan hasil nilai keuntungan yang akan di tampilkan pada program tersebut.

```
for(int i=2;i<11;i++){
    //j = 2 karna dari row ke 3 , j<7 karna sampai row 6
    for(int j=2;j<7;j++){
        valzj=valzj+(Float.valueOf(jTable1.getValueAt(j, i).toString())*Float.valueOf(jTable1.getValueAt(j, 0).toString()));
    }
    jTable1.setValueAt(valzj, 7, i); //untuk mengisi zj
    valzjcj=(valzj-Float.valueOf(jTable1.getValueAt(0, i).toString()));
    jTable1.setValueAt(valzjcj, 8, i);
    if(valzjcj>zjmax){
        zjmax=valzjcj;
        pivot=i-1;
    }
    valzj=0;
}
```

**Gambar 4.6** Perhitungan  $z_j$  dan  $c_j - z_j$

Perhitungan untuk  $z_j$  dan  $c_j - z_j$  pada semua iterasi saat  $c_j - z_j$  sampai nol atau minus nilainya

```
private int cekhitung(){
    int cek=0;
    float valu=0;

    for(int j=2;j<11;j++){
        valu=Float.valueOf(jTable1.getValueAt(8, j).toString());

        if(valu>0){
            cek=1;
        }
    }

    return cek;
}
```

**Gambar 4.7** Menentukan iterasi dilanjutkan atau tidak

Untuk menentukan  $z_j - c_j$  yang lebih dari nol akan lanjut untuk menghitung rasio sampai kurang dari nol

```

if(cek==1){
    float rasio=0;
    float rasiomax=999999999;
    int pivott=0;
    for(int i=2;i<7;i++){
        rasio=Float.valueOf(jTable1.getValueAt(i, 11).toString())/Float.valueOf(jTable1.getValueAt(i, pivott+1).toString());
        jTable1.setValueAt(rasio, i, 12);
        if(rasio<rasiomax){
            if(index==1641-1==Integer.parseInt(tablesave[index-1][8][11])){
            }
            else{
                rasiomax=rasio;
                pivott=i-1;
            }
        }
    }
}

jTable1.setValueAt(pivot, 8, 11); //untuk menyimpan pivot kolom
jTable1.setValueAt(pivott, 8, 12); //untuk menyimpan pivot baris
}
return cek;

```

**Gambar 4.8** Mencari kolom kunci dan baris kunci

Untuk menghitung rasio terkecil dan untuk mencari  $z_j - c_j$  max untuk mendapatkan kolom kunci dan baris kunci *pivot*.

```

while(cek==1){
    idxs=idxs+1;
    if(idxs==1){
        //untuk iterasi 1
        newiteawal(Integer.valueOf(tablesave[idxs-1][8][11]), Integer.valueOf(tablesave[idxs-1][8][12]));
        cek=hitung(idxs);
        save(idxs);
    }
    else{
        newite(Integer.valueOf(tablesave[idxs-1][8][11]), Integer.valueOf(tablesave[idxs-1][8][12]), idxs);
        cek=hitung(idxs);
        System.out.println(idxs);
        save(idxs);
    }
}

```

**Gambar 4.9** Menghitung iterasi

Untuk menghitung proses iterasi sampai selesai, proses pertama newiteawal mrnggunakan method itu, proses ke dua menggunakan method newite untuk menjalankan proses selanjutnya

```

for(int i=0;i<5;i++){
    for(int j=0;j<10;j++){
        if(i==pivott-1){
            //ini untuk baris pivot iterasi pertama
            hnewval[i][j]=Float.valueOf(jTable1.getValueAt(i+2, j+2).toString());
        }
        else{
            float a,b,c;
            a=Float.valueOf(jTable1.getValueAt(i+2, j+2).toString());
            b=Float.valueOf(jTable1.getValueAt(i+2, pivott+1).toString());
            c=Float.valueOf(jTable1.getValueAt(pivott+1, j+2).toString());
            hnewval[i][j]=a-(b*c);
        }
        // ini untuk perhitungan lainnya
    }
}

```

**Gambar 4.10** Tampilan iterasi baru

Tampilan source code perhitungan iterasi pertama, untuk baris pivot iterasi pertama di dapatkan dari baris kunci tabel sebelumnya, untuk perhitungan  $a - (b * c)$  dengan cara dikali dulu trus dikurangi

```
private void newite(int pivota, int pivotb, int index) {
    float[][] hnewval = new float[5][10];
    String[][] cjs = new String[5][2];
    //hitung nilai pada baris pivot
    System.out.println(" "+pivota+pivotb+"\n");
    for(int j=0; j<10; j++) {
        float a, b;

        a = Float.valueOf(tablesave[index-1][pivotb+1][j+2]);
        b = Float.valueOf(tablesave[index-1][pivotb+1][pivota+1]);
        hnewval[pivotb-1][j] = a/b;

    }
    //hitung sisanya
}
```

**Gambar 4.11** Tampilan mencari selain iterasi pertama pada baris kunci

Tampilan untuk menjelaskan perhitungan iterasi selanjutnya pada baris kunci dengan cara  $a / b$  membuat jadi satu elemen kuncinya, yang satu baris dengan elemen kunci harus dioperasikan sama.

```
for(int i=0; i<5; i++) {
    for(int j=0; j<10; j++) {
        if(i!=pivotb-1 && Float.valueOf(tablesave[index-1][i+2][pivota+1])>0) {
            float a, b, c;
            a = Float.valueOf(tablesave[index-1][i+2][j+2]);
            b = Float.valueOf(tablesave[index-1][i+2][pivota+1]);
            c = hnewval[pivotb-1][j];
            hnewval[i][j] = a - (b*c);
        }
        else if(i!=pivotb-1) {
            float a, b;
            a = Float.valueOf(tablesave[index-1][i+2][j+2]);
            b = hnewval[pivotb-1][j];
            hnewval[i][j] = a + b;
        }
    }
}
```

**Gambar 4.12** Tampilan mencari selain iterasi pertama

Tampilan untuk menjelaskan perhitungan iterasi selain pada baris kunci dengan cara  $a + b$  elemen yang satu kolom dengan pivot itu negatif maka ditambahkan supaya jadi nol.

The screenshot shows the NetBeans IDE interface. A CSV file named 'inputpg.csv' is open, displaying a table of numerical data. Below the table, there are controls for 'Pivot Baris' and 'Pivot Kolom', both set to 3. A message dialog box is displayed in the foreground, showing the calculation:  $\text{Keuntungan} = (545 \times 2273) + (943 \times 682) = 1881911$ .

	cj	0	0	4	0	3	0	0	1000	0		
cj	xj/bj	x1	x2	s1-	s1+	s2-	s2+	s3	r1	s4	rhs	rasio
4	s1-	0.0	0.0	1.0	-1.0	0.0	0.0	0.0	2.324...	-56.2...	3184...	
0	x2	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.007...	-0.04...	681.8...	
0	s3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	34.78...	-383...	1549...	
0	s2+	0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.007...	-0.04...	631.8...	
0	x1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.01...	0.181...	2272...	
0	zj	0.0	0.0	4.0	-4.0	0.0	0.0	0.0	9.297...	-224...		
0	zj-cj	0.0	0.0	0.0	-4.0	-3.0	0.0	0.0	-990...	-224...		

**Gambar 4.13** Tampilan Hasil akhir dari pengoptimalan

Tampilan hasil akhir menggunakan goal programming dari  $x_1 545 \times 2.273 + x_2 943 \times 682 = 1.881.991$  untuk mencari keuntungan.

Hasil dari pengujian pada sistem ini adalah membandingkan keuntungan sebelum dan sesudah melakukan proses optimasi menggunakan Goal Programming, perbandingan bisa dilihat pada tabel berikut ini :

**Tabel 4.1** perbandingan keuntungan antara sebelum dan sesudah optimasi menggunakan Goal Programming.

$x_i$	Sebelum Optimal	Sesudah Optimal
Roti Gulung	2.500	2.273
Donat	500	681
Keuntungan	1.834.000	1.881.911

Hasil dari pengoptimalan antara roti gulung dan donat sesudah di lakukan pengujian sistem di dapatkan hasil dari  $x_1 2.273$  dan  $x_2 681$  yang di hitung mendapatkan total keuntungan 1.881.911 yang sebelumnya 1.834.000.

## HASIL DAN PEMBAHASAN

### KESIMPULAN

Pada penelitian yang telah dilakukan dengan menggunakan metode goal programming dapat disimpulkan bahwa hasil perhitungan keuntungan penjualan roti gulung dan pendapatan penjualan donat pada Toko Roti Tiga Bintang adalah dengan memproduksi  $x_1$  sebanyak 2273 dan  $x_2$  sebanyak 682 sehingga optimasi perencanaan produksi pada Toko Roti Tiga Bintang dengan kendala-kendala sasaran yang tersedia sehingga menghasilkan tujuan-tujuan yang optimum yaitu memaksimalkan pendapatan sebesar 1.881.911 yang sebelumnya sebesar 1.834.000.

### UCAPAN TERIMA KASIH

Pada bagian ini bersifat optional, boleh dihilangkan oleh penulis. Ucapan terima kasih berisikan prakata apresiasi penulis kepada orang, kelompok atau instansi yang berkontribusi pada program penelitian.

### DAFTAR PUSTAKA

- [1] Center for History and New Media, "Zotero Quick Start Guide." [Online]. Available: [http://zotero.org/support/quick\\_start\\_guide](http://zotero.org/support/quick_start_guide).
- [2] E. J. Marmel, *Word 2016*. Indianapolis, IN: Visual, an imprint of Wiley, John Wiley & Sons, Inc, 2016.
- [3] D. Graffox, "IEEE Citation Reference." Sep-2009.
- [4] M. Nur *et al.*, "Evaluation of Novel Integrated Dielectric Barrier Discharge Plasma as Ozone Generator," *Bull. Chem. React. Eng. Catal.*, vol. 12, no. 1, p. 24, Apr. 2017.
- [5] P. N. Kementerian, "Permendiknas No. 17 tahun 2010 tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi," 2010. [Online]. Available: <http://idr.iain-antasari.ac.id/479/>. [Accessed: 11-Jun-2017].
- [6] R. Macfarlane, *Original Copy: Plagiarism and Originality in Nineteenth-Century Literature*, 1 edition. Oxford ; New York: Oxford University Press, 2007.
- [7] D. Gookin, *Word 2016 for professionals for dummies*. Hoboken, NJ: John Wiley & Sons, 2016.