OPTIMALISASI PRODUKSI ROTI DENGAN MENGGUNAKAN METODE *BRANCH AND BOUND*

(Studi Kasus: Roti Kasur, Roti Kepang, Roti Bantal dan Roti Sobek pada Safa Bakery Kecamatan Kayen Pati)

SKRIPSI

Diajukan untuk Memenuhi Sebagian Syarat Guna

Memperoleh Gelar Sarjana Sains

dalam Ilmu Matematika

Oleh:

NUNU MARWATI

NIM: 1708046019

PROGRAM STUDI MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI WALISONGO

SEMARANG

2023

PERNYATAAN KEASLIAN

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Nunu Marwati

NIM : 1708046019

Jurusan : Matematika

Menyatakan bahwa skripsi yang berjudul:

OPTIMALISASI PRODUKSI ROTI DENGAN MENGGUNAKAN METODE *BRANCH AND BOUND* (Studi Kasus: Roti Kasur, Roti Kepang, Roti Bantal dan Roti Sobek pada Safa Bakery Kecamatan Kayen Pati)

Secara keseluruhan adalah hasil penelitian atau karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 28 Juni 2023

Pembuat Pernyataan,

METERAI TEMPEL

25DAKX504003496

Nunu Marwati

NIM: 1708046019

PENGESAHAN

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
FAKULTAS SAINS DAN TEKNOLOGI
JL. Prof. Dr. H. Hamka Ngaliyan Semarang 50185.
Telp. 024-7601295 Fax.7615387

PENGESAHAN

Naskah skripsi berikut ini :

Judul	: **Optimalisasi Produkai Roti dengan Menggunakan Metode *Branch and Bound* (Studi Kasus: Roti Kasur, Roti Kepang, Roti Bantal dan Roti Sobek pada Safa Bakery Kecamatan Kayen Pati)**
Nama	: Nunu Marwati
NIM	: 1708046019
Jurusan	: Matematika

Telah diujikan dalam sidang tugas akhir oleh Dewan Penguji Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang dan dapat diterima sebagai syarat memperoleh gelar Sarjana dalam Ilmu Matematika.

Semarang, 28 Juni 2023

Ketua Sidang	Sekretaris Sidang
Prihadi Kurniawan, M.Sc.	**Yolanda Norasia, M.Si.**
NIP. 199012262019031012	NIP.199409232019032011
Penguji Utama I	Penguji Utama II
Any Muanalifah, M.Si, P.hd	**Aini Fitriyah, S.Pd, M.Sc.**
NIP. 198201132011012009	NIP. 198909292019032021
Pembimbing I	Pembimbing II
Siti Maslihah, M.Si.	**Zulaikha, M.Si.**
NIP.197706112011012004	NIP.199204092019032027

KEMENTERIAN AGAMA
FAKULTAS SAINS DAN TEKNOLOGI
UIN WALISONGO
SEMARANG
REPUBLIK INDONESIA

NOTA PEMBIMBING

NOTA PEMBIMBING

Semarang, 24 Mei 2023

Yth. Ketua Program Studi Matematika
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum wr.wb

Dengan ini diberitahukan bahwa, saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan:

Judul : **Optimalisasi Produksi Roti dengan Menggunakan Metode *Branch and Bound* (Studi Kasus: Roti Kasur, Roti Kepang, Roti Bantal dan Roti Sobek pada Safa Bakery Kecamatan Kayen Pati)**

Nama : **Nunu Marwati**

NIM : 1708046019

Jurusan: Matematika

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diujikan dalam sidang Munaqasyah.

Wassalamu'alaikum wr.wb

Pembimbing I,

Siti Maslihah, M. Si
NIP. 19770611 201101 2 004

NOTA PEMBIMBING

NOTA PEMBIMBING

Semarang, 16 Juni 2023

Yth. Ketua Program Studi Matematika

Fakultas Sains dan Teknologi

UIN Walisongo Semarang

Assalamu'alaikum wr.wb

Dengan ini diberitahukan bahwa, saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan:

Judul : **Optimalisasi Produksi Roti dengan Menggunakan Metode *Branch and Bound* (Studi Kasus: Roti Kasur, Roti Kepang, Roti Bantal dan Roti Sobek pada Safa Bakery Kecamatan Kayen Pati)**

Nama : **Nunu Marwati**

NIM : 1708046019

Jurusan: Matematika

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diujikan dalam sidang Munaqasyah.

Wassalamu'alaikum wr.wb

Pembimbing II,

Zulaikha, M. Si

NIP: 19920409 201903 2 027

MOTTO

“Tidak ada yang tidak mungkin, bagi dia yang mau berusaha. Setiap detik adalah kesempatan untuk mengubah hidup. *Hari ini lebih baik dari hari kemarin, hari esok lebih baik dari hari ini dan iringilah perjuanganmu dengan keyakinan*”.

ABSTRAK

Pertumbuhan produksi roti sebagai salah satu UKM (usaha kecil menengah) yang berkembang di tengah masyarakat menunjukkan bahwa usaha roti masih dapat terus berkembang dan merupakan salah satu pasar potensial untuk mencapai keuntungan optimal. Safa Bakery merupakan salah satu usaha dagang yang bergerak dalam bidang pembuatan roti di daerah Pati. Usaha dagang ini memproduksi roti dengan berbagai jenis rasa. Dalam penelitian ini diambil 4 jenis roti, yaitu Roti Kasur, Roti Kepang, Roti Bantal dan Roti Sobek. Masalah mengoptimalkan jumlah produksi akan dimodelkan ke dalam bentuk model matematika berupa program linier, kemudian dilanjutkan dengan program *integer*, dimana variabel keputusan harus berupa bilangan bulat (*integer*). Masalah program *integer* tersebut akan diselesaikan dengan menggunakan metode *Branch and Bound* yang terlebih dahulu menghitung nilai variabel keputusan dengan menggunakan metode simpleks. Metode *Branch and Bound* merupakan metode yang digunakan untuk menyelesaikan masalah program linier yang menghasilkan penyelesaian dalam bentuk bilangan bulat (*integer*). Dalam penelitian ini penulis menggunakan bantuan POM for Windows dan MATLAB untuk menyelesaikan masalah program linier. Analisis metode *Branch and Bound* diperoleh selisih nilai keuntungan penjualan sebesar 2,79% atau senilai Rp 402.800,00 dari keuntungan Safa Bakery. Jumlah roti yang diproduksi dari bahan-bahan yang tersedia adalah 5.500 roti dimana Roti Kasur diproduksi sebanyak 1.282 roti, Roti Kepang diproduksi 966 roti, Roti Bantal diproduksi 1.500 roti dan Roti Sobek diproduksi sebanyak 1.752 roti dengan keuntungan sebesar Rp 14.802.400,00.

Kata Kunci: Program Linier, Program *Integer*, Metode Simpleks, Metode *Branch and Bound,* POM for Windows, MATLAB.

TRANSLITERASI ARAB-LATIN

Penulisan transliterasi huruf-huruf Arab latin dalam skripsi ini berpedoman pada SKB Menteri Agama dan Menteri Pendidikan dan Kebudayaan R.I. Nomor : 158/1987 dan Nomor : 0543/u/1987. Penyimpangan penulisan kata sandang [al-] disengaja secara konsisten supaya sesuai teks Arabnya.

ا	A	ط	t}
ب	B	ظ	z}
ت	T	ع	‘
ث	s\	غ	G
ج	J	ف	F
ح	h}	ق	Q
خ	kh}	ك	K
د	D	ل	L
ذ	z\	م	M
ر	R	ن	N
ز	Z	و	W
س	S	ه	H
ش	Sy	ء	’
ص	s}	ى	Y
ض	d}		

Bacaan Madd :
a >= a panjang
i > = i panjang
u >=u panjang

Bacaan Diftong :
au = أو
ai = أى
iy = إى

KATA PENGANTAR

Puji syukur Alhamdulillah penulis panjatkan kepada Allah SWT atas segala rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan skripsi ini dalam rangka memenuhi syarat guna memperoleh gelar sarjana Strata Satu (S1) Prodi Matematika UIN Walisongo Semarang. Shalawat serta salam penulis curahkan kepada Nabi Muhammad SAW yang telah membawa risalah-Nya, sehingga dapat menjadi bekal hidup di dunia dan di akhirat kelak.

Ucapan terimakasih penulis sampaikan kepada semua pihak yang telah memberikan bimbingan, arahan, saran, dan motivasi sehingga skripsi ini dapat penulis selesaikan dengan baik. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan mengucapkan terimakasih kepada:

1. Allah SWT yang telah memberikan rahmat, karunia, hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini
2. Prof. Dr. H. Imam Taufiq, M.Ag., selaku rektor Universitas Islam Negeri Walisongo Semarang
3. Bapak Dr. H. Ismail, M.Ag., selaku ketua Dekan Fakultas Sains dan Teknologi UIN Walisongo Semarang
4. Ibu Emy Siswanah, M.Sc., selaku Ketua Program Studi Matematika UIN Walisongo Semarang

5. Bapak Aunur, M.Pd., selaku Sekertaris Program Studi Matematika UIN Walisongo Semarang
6. Ibu Aini Fitriyah, S.Pd., M.Sc., selaku wali dosen penulis
7. Ibu Siti Maslihah, M.Si., selaku dosen pembimbing I yang telah memberikan waktu dan bimbingannya sehingga skripsi ini dapat selesai
8. Ibu Zulaikha, M.Si., selaku dosen pembimbing II yang telah memberikan waktu dan bimbingannya sehingga skripsi ini dapat selesai
9. Segenap dosen, pegawai, dan seluruh civitas akademik lingkungan UIN Walisongo Semarang khususnya para dosen jurusan Matematika
10. Keluarga besar Safa Bakery yang telah memberikan bimbingannya pada saat penelitian
11. Kedua orang tua penulis, Bapak Rejo dan Ibu Martuni yang selalu memberikan semangat, dukungan dan yang terpenting do'a restu kepada penulis untuk mencapai kesuksesan
12. Mas Andi Kumala, Budhe Narsi, Mbak Sri Lestari dan keluarga besar yang selalu memberikan cinta kasih, motivasi, dukungan, semangat, serta doa kepada penulis
13. Bapak Ahmad Amnan Muqoddam, Ibu Rofiqotul Makiyyah, Ibu Muhimmah Thoyfoer beserta keluarga ndalem yang memberikan ilmu dan bimbingannya kepada penulis

14. Teman-teman Matematika 2017 atas kebersamaan selama menjalani proses perkuliahan
15. Seluruh pihak yang telah membantu penyelesaian skripsi yang tidak dapat penulis sebutkan satu persatu. Semoga Allah SWT memberikan anugerah dan barokah-Nya kepada semua pihak yang telah membantu penulis menyelesaikan skripsi ini.

Penulis menyadari bahwa dalam penulisan skripsi ini masih banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa yang akan datang. Semoga skripsi ini dapat bermanfaat bagi pembaca. Amiin Yaa Rabbal'Alamiin

Semarang, 28 Juni 2023
Penulis

Nunu Marwati
NIM. 1708046019

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

A. Latar Belakang

Pemanfaatan pengetahuan tentang riset operasi dalam kehidupan sehari-hari telah meluas dan umum digunakan oleh manusia, seperti dalam konteks ekonomi. Industri makanan, seperti roti, merupakan salah satu yang memainkan peranan penting dalam pertumbuhan ekonomi. Hal ini terbukti dengan meningkatnya minat masyarakat terhadap roti. Dampaknya, bisnis bakery menjadi semakin berkembang dan menjadi pasar yang sektor menjanjikan untuk meraih keuntungan yang besar.

Roti adalah salah satu jenis pangan yang lumayan digemari sebagai bahan pangan selain nasi. Roti tersusun atas komponen dasar gandum atau padi, sebagai produk turunan hasil pertanian. Produk pertanian utama yang berkontribusi terhadap ketersediaan pangan bagi masyarakat adalah padi. Ini menunjukkan bahwa semakin banyak orang mengonsumsi roti daripada makanan olahan lain yang terbuat dari tepung terigu (Jumarin, 2012). Pertumbuhan pesat dalam bidang produksi dan peningkatan populasi manusia yang terus meningkat mengakibatkan peningkatan jumlah produksi barang sesuai dengan kebutuhan.

Selain fungsi lainnya, produksi merupakan faktor kunci keberhasilan suatu perusahaan. Fenomenanya adalah produksi menawarkan persaingan yang semakin ketat, dengan berbagai keunggulan tergantung pada teknologi, kualitas atau biaya serta daya saing lainnya. Selain itu, inovasi dan kemampuan mereka untuk menjawab permintaan masyarakat. Produksi produk yang memberikan kontribusi besar terhadap keuntungan adalah salah satu usahanya (Herman, 2008).

Ada banyak keuntungan yang bisa didapat dari modal kecil, sehingga muncul masalah optimasi. Semua perusahaan akan berusaha memaksimalkan profitabilitas atau mengurangi biaya, dengan maksud untuk mencapai kondisi yang optimal. Setiap perusahaan akan berusaha untuk memaksimalkan keuntungan atau meminimalkan biaya yang dikeluarkan untuk mencapai kondisi yang optimal. Optimasi adalah cara memaksimalkan keuntungan. Namun, untuk membelanjakan uang paling sedikit, minimalisasi adalah pengoptimalan. Proses optimalisasi melibatkan pengelolaan biaya dan upaya maksimal dalam memanfaatkan sumber daya yang tersedia untuk mencapai hasil yang optimal. Tujuan dari optimalisasi adalah untuk memaksimalkan atau meminimalkan suatu fungsi tertentu dengan mempertimbangkan sejumlah batasan, seperti mencapai pengembalian maksimum atau biaya minimum (Karo, 2016).

Optimasi melibatkan penggunaan model matematika untuk memilih solusi yang paling tepat untuk suatu masalah. Program linier mampu memecahkan masalah.

Pemrograman linier adalah metode yang digunakan untuk mencari solusi optimal dalam masalah pengambilan keputusan dengan merumuskan fungsi tujuan (maksimisasi atau minimisasi) dan batasan sebagai model matematika berupa persamaan linier. Pemrograman linier umumnya digunakan untuk mengatasi masalah alokasi sumber daya dengan efisien (Sitorus, 1997). Teknik ini sering digunakan dalam berbagai jenis masalah, termasuk masalah transportasi, penugasan, program dinamis, dan program bilangan bulat. Dengan menggunakan pemrograman linier, banyak masalah tersebut dapat diselesaikan dengan solusi yang optimal.

Pemrograman linier bilangan bulat adalah suatu pendekatan dalam pemrograman linier yang khusus digunakan untuk menyesuaikan masalah di mana solusi optimal harus memiliki nilai variabel keputusan yang merupakan bilangan bulat. Alasan untuk membatasi variabel keputusan menjadi bilangan bulat adalah karena dalam beberapa konteks, seperti roti, rumah, sepatu, dan lain sebagainya, nilai variabel tersebut tidak dapat berupa pecahan (Sitorus, 1997). Pemrograman linier bilangan bulat dapat diselesaikan menggunakan berbagai metode, termasuk grafik, metode substitusi dan eliminasi, dan lain-lain. Namun, metode yang efisien untuk menyelesaikan masalah pemrograman linier bilangan bulat adalah dengan menggunakan metode *Branch and Bound* dibandingkan dengan metode komputasi bilangan bulat lainnya, dan metode ini telah menjadi standar dalam kode komputer untuk pemrograman linier bilangan bulat (Pagiling, Sahari & Rais, 2015).

Metode *Branch and Bound* telah menjadi pilihan yang umum untuk menyelesaikan masalah pemrograman bilangan bulat karena dapat menghasilkan solusi optimasi yang lebih akurat dan baik dibandingkan dengan metode lainnya. Kelebihan utama metode ini adalah kemampuannya untuk menemukan lebih dari satu solusi optimal, sehingga memungkinkan pemilihan solusi terbaik dari hasil yang diperoleh (Supatimah, dkk, 2019). Metode *Branch and Bound* telah digunakan secara luas dalam konteks pemrograman linier dengan variabel keputusan bilangan bulat. Prinsip dasar dari metode *Branch and Bound* adalah dengan melakukan percabangan pada variabel-variabel keputusan yang tidak menghasilkan solusi bilangan bulat, dan percabangan akan terus dilakukan hingga ditemukan resolusi di mana semua keputusan memiliki nilai bilangan bulat, yang menghasilkan keuntungan maksimal bagi perusahaan. Salah satu contohnya adalah dalam optimisasi jumlah produksi roti di mana nilai optimalnya harus berupa bilangan bulat.

Metode *Branch and Bound* telah banyak diteliti sebelumnya oleh beberapa peneliti. Misalnya, Pagiling (2015) dalam penelitiannya yang berjudul *"Optimalisasi Hasil Produksi Tahu dan Tempe Menggunakan Metode Branch and Bound (Studi Kasus: Pabrik Tempe Eri JL. Teratai No. 04 Palu Selatan)"* menjelaskan tentang penerapan Metode *Branch and Bound* dalam mengoptimalkan hasil produksi tahu dan tempe dengan mempertimbangkan berbagai batasan yang terkait dengan bahan baku. Suryawan, dkk (2016) juga menjelaskan dalam penelitian mereka yang berjudul *"Penerapan Branch and Bound Algorithm dalam Optimalisasi Produksi Roti"* tentang cara mengoptimalkan jumlah produksi roti di Ramadhan Bakery untuk mencapai keuntungan maksimal dengan memperhatikan batasan bahan baku. Selain itu, Nurjanah (2018) dalam penelitiannya yang berjudul *"Metode Branch and Bound untuk Meminimalkan Biaya Bahan Baku"* menjelaskan tentang penggunaan Metode *Branch and Bound* dalam meminimalkan biaya bahan baku yang digunakan dalam pembuatan kue di Home Industry Bunda Bakery di jalan Bakti Permai.

Berdasarkan rujukan penelitian di atas penulis tertarik untuk meneliti tentang "Optimalisasi Produksi Roti dengan Menggunakan Metode *Branch and Bound (*Studi

Kasus: Roti Kasur, Roti Kepang, Roti Bantal dan Roti Sobek pada Safa Bakery Kecamatan Kayen Pati*)".*

B. Rumusan Masalah

1. Bagaiman cara menggunakan metode *Branch and Bound* untuk mencari kombinasi jumlah roti yang optimal untuk keuntungan maksimal?
2. Bagaimana perbandingan keuntungan yang diperoleh sesudah dan sebelum dilakukan penelitian dengan menggunakan metode *Branch and Bound?*

C. Batasan Masalah

Dengan dilakukannya penelitian maka dapat dilakukan pencarian yang fokus dan tepat, dengan diberikan batasan masalah sebagai berikut:

1. Empat jenis produk roti yaitu Roti Kasur, Roti Kepang, Roti Bantal, dan Roti Sobek
2. Proses produksi berjalan lancar tanpa gangguan
3. Pada akhir produksi, harga bahan baku diasumsikan konstan. (Biaya bahan baku yang dianggap konstan adalah biaya perolehan bahan baku yang tidak berubah selama proses penelitian)

D. Tujuan Penelitian

Tujuan utama dari penelitian ini adalah untuk menunjukkan bahwa, sebagai alternatif untuk mengoptimalkan produksi roti, teknik *Branch and Bound* dalam pemrograman linier dapat digunakan.

E. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi Usaha Dagang

   Untuk tujuan memaksimalkan keuntungan produksi dan sebagai salah satu alternatif untuk mencapai keuntungan yang tinggi, maka penelitian ini dapat diperhitungkan baik dengan cara masukan maupun pertimbangan.

2. Bagi Peneliti

   Peneliti mendapatkan pengalaman dalam mengoptimalkan produksi dengan menerapkan ilmu yang didapat melalui metode *Branch and Bound.*

3. Bagi Universitas

   Menambah literatur universitas saat ini, terutama dalam hal optimalisasi proses produksi melalui pendekatan pemrograman yang sistematis.

BAB II
TINJAUAN PUSTAKA

A. Program Linier

Pada tahun 1947, George B. Dantzig menemukan teknik pemrograman linier yang merupakan metode penerapan matematika. Konsep "linier" dalam hal ini mengacu pada konsistensi fungsi persamaan matematika atau pertidaksamaan yang terdapat dalam permasalahan tersebut. Pemrograman, dalam konteks ini, memiliki keterkaitan dengan model perencanaan. Ini berarti bahwa kegiatan perencanaan dilakukan untuk mencapai hasil optimal, yaitu hasil yang paling sesuai dengan tujuan atau sasaran yang ditentukan, dan menjadi bagian integral dari program secara keseluruhan. Dengan demikian, pemrograman linier digunakan sebagai dasar dalam pengambilan keputusan terkait masalah nyata yang diwujudkan atau dimodelkan. Dalam pemrograman linier, alam semesta atau sistem yang sedang dipelajari digambarkan sebagai model matematika yang mencakup fungsi tujuan yang ingin dicapai.

Teknik atau metode pemrograman linier menggunakan pendekatan matematika untuk menyelesaikan masalah alokasi sumber daya yang terbatas di antara berbagai aktivitas. Tujuan utamanya adalah untuk memaksimalkan keuntungan atau mengurangi biaya, dengan mempertimbangkan batasan yang ada. Dalam istilah yang lebih sederhana, pemrograman linier dapat dijelaskan sebagai teknik pengoptimalan yang melibatkan sistem linier yang terdiri dari fungsi tujuan dan kendala-kendala yang membatasi solusi. Penelitian oleh Rafflesia dan Widodo pada tahun 2014 memberikan pemahaman yang lebih rinci tentang metode ini dan penerapannya dalam konteks pengalokasian sumber daya yang terbatas.

Konsep utama dalam pemrograman linier adalah memastikan bahwa masalah didefinisikan dengan jelas dan menggunakan informasi yang cukup. Setelah itu, langkah selanjutnya adalah mentransformasikan masalah tersebut menjadi model matematika yang lebih terstruktur dan dapat digunakan untuk mencari solusi yang lebih baik (Siagian, 1987). Pemrograman linier dianggap bermasalah jika kondisi berikut terpenuhi:

a. Tujuan yang ingin dicapai harus dirumuskan dalam bentuk fungsi linier yang dikenal sebagai fungsi tujuan atau *objective function*. Tujuan tersebut dapat berupa maksimisasi keuntungan, minimisasi biaya, dan sejenisnya.
b. Solusi harus ditemukan. Pemrograman linier bertujuan untuk mencari solusi yang mengoptimalkan

fungsi tujuan, seperti mencari keuntungan maksimum atau biaya minimum.

c. Terdapat keterbatasan pada sumber daya yang tersedia, seperti ketersediaan bahan baku yang terbatas, kapasitas model yang terbatas, ruang penyimpanan barang yang terbatas, dan sebagainya. Dalam kasus pertidaksamaan linier, kendala-kendala tersebut harus dinyatakan.

1. Model Program Linier

   Ada karakteristik masalah berikut yang dapat diselesaikan dengan model linier:

   a. Tidak ada variabel penyusun negatif.
   b. Nilai objektif dapat dinyatakan sebagai fungsi linier dari variabel.
   c. Sistem persamaan linier dapat digunakan untuk menyatakan kendala (Rafflesia, 2004)

   Pemrograman linier yang merupakan metodologi analisis yang menggunakan model matematika untuk mengidentifikasi opsi yang berbeda untuk memecahkan masalah, bertujuan untuk mengidentifikasi beberapa solusi yang paling sesuai. Pemrograman linier memiliki bentuk umum, sebagai berikut:

Fungsi Tujuan:

Maksimumkan atau minimumkan

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

Dengan kendala yang membatasi:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$
$$x_1, x_2, \dots, x_n \geq 0$$

Keterangan:

x_n = Menunjukkan variabel keputusan

c_n = Koefisien fungsi tujuan

a_{mn} = Koefisien fungsi kendala

b_m = Jumlah tiap kendala yang ada

Karakteristik yang paling umum digunakan dalam masalah program linier digunakan untuk kontruksi model berdasarkan rumusan masalah sebagai berikut ini:

1. Variabel Keputusan

   Variabel keputusan merujuk pada variabel yang menggambarkan keputusan yang akan diambil dalam masalah pemrograman linier. Variabel ini memberikan informasi detail tentang keputusan yang akan dilakukan.

2. Fungsi Tujuan

Fungsi tujuan adalah hubungan linier antara variabel keputusan yang menggambarkan tujuan yang ingin dicapai dalam masalah pemrograman linier. Fungsi tujuan ini dapat berupa fungsi maksimalisasi atau minimalisasi, yang bergantung pada tujuan yang ingin dicapai. Fungsi tujuan ini dibatasi oleh kendala yang mencerminkan keterbatasan kapasitas, waktu produksi, atau keterampilan yang tersedia.

3. Pembatas

Pembatasan adalah kendala-kendala yang membatasi nilai variabel keputusan dalam mencapai keseimbangan yang diinginkan. Koefisien variabel keputusan dalam setiap pembatasan disebut koefisien teknologi, sementara nilai yang ada di sisi kanan setiap pembatasan disebut sisi kanan pembatasan. Pembatasan ini mewakili keterbatasan yang harus dipenuhi dalam memecahkan masalah pemrograman linier.

4. Pembatas Tanda

Pembatas tanda dalam pemrograman linier menentukan apakah variabel keputusan hanya boleh memiliki nilai non-negatif atau dapat memiliki nilai positif dan negatif (tidak terbatas pada tanda) (Dimyati, 2006).

2. Metode Grafik

Batasan grafik digunakan sebagai alat untuk menentukan titik yang paling efektif dalam metode grafik ini. Kendala dalam sistem program linier akan selalu membentuk bidang sisi-n yang merupakan himpunan konveks, dimana titik optimum harus terletak pada sudut bidang yang terbentuk. Hal ini cukup mudah dilakukan dengan menggunakan metode grafik, meskipun membatasi diri pada dua kendala. Dimensi ruang harus didefinisikan dengan batasan dalam metode *Linear Programming*. Artinya dengan jumlah kendala lebih dari 4 maka permasalahan tidak dapat dideskripsikan secara grafis sehingga metode grafik tidak dapat digunakan. Meskipun masalah dengan 3 batasan sangat sulit untuk ditentukan karena gambar digambar dalam 3 dimensi dan sulit untuk bekerja dengan metode grafik karena alasan tersebut.

Langkah-langkah penyelesaian menggunakan metode grafik adalah sebagai berikut:

a. Tentukan model masalah yang dihadapi.
b. Serangkaian gambar grafik yang berkaitan dengan kendala.
c. Mengidentifikasi daerah yang layak, yaitu tempat pada graf yang memenuhi semua batasan.

d. Menghitung nilai suatu fungsi pada titik sudut-n dimensi pada suatu daerah yang layak.
e. Tentukan titik yang paling memberikan nilai fungsi optimal (Listyaningrum dan Afiati, 2018).

3. Metode Simpleks

Metode simpleks, yang dikembangkan oleh George Dantzig pada tahun 1947, adalah pendekatan sistematis yang mengiterasi dari solusi dasar yang layak ke solusi dasar lainnya dengan jumlah pengulangan terbatas. Metode ini bertujuan mencapai solusi dasar optimal akhir dengan memperbaiki fungsi tujuan pada setiap langkah, sehingga selalu menghasilkan solusi yang lebih baik atau mendekati langkah sebelumnya. Berbeda dengan pemrograman linier dan metode grafik yang terbatas pada kasus dengan dua variabel keputusan, metode simpleks dapat digunakan untuk menyelesaikan kasus dengan jumlah variabel keputusan yang tidak terbatas (Purba dan Ahyaningsing, 2020).

Metode grafik adalah metode yang dapat digunakan untuk menyelesaikan program linier dengan dua variabel keputusan. Namun, untuk masalah pemrograman linier yang kompleks, metode grafik menjadi tidak efisien dan diperlukan metode aljabar yang lebih lanjut. Salah satu metode aljabar yang paling umum digunakan adalah metode simpleks, yang melibatkan iterasi berulang hingga mencapai solusi yang diinginkan. Dengan menggunakan metode simpleks, masalah yang sulit dapat dipecahkan dengan menggantikannya dengan masalah yang lebih mudah yang dapat dipecahkan secara aljabar (Akram, A. Sahari, A. I. Jaya, 2016).

Langkah-langkah penanganan program linier dalam metode simpleks adalah sebagai berikut:

1. Merumuskan dan mengkonstruksikan masalah program linier dalam bentuk standar.
2. Buatlah tabel awal simpleks, seperti pada Tabel 2.1 berikut:

Tabel 2.1. Tabel Awal Simpleks

		c_j	c_1	c_2	...	c_n	0	0	...	0
Variabel Dasar	Tujuan	x_j / b_i	x_1	x_2	...	x_n	s_1	s_2	...	s_m
s_1	0	b_1	a_{11}	a_{12}	...	a_{1n}	1	0	...	0
s_2	0	b_2	a_{21}	a_{22}	...	a_{2n}	0	1	...	0
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
s_m	0	b_m	a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1
	z_j	0	0	0	...	0	0	0	...	0
		$c_j - z_j$	c_1	c_2	...	c_n	0	0	...	0

Keterangan:

a. Pada kolom pertama terdapat variabel dasar yang termasuk variabel *slack* yang akan menggantikan variabel keputusan.
b. Kolom ketiga berisi konstanta c_j yang ada di sisi kanan setiap kendala atau konstanta fungsi kendala.
c. Baris c_j mengandung koefisien biaya relatif dari fungsi tujuan dan kolom variabel slack dengan nilai nol.
d. Variabel keputusan dan variabel *slack* dimasukkan dalam baris a_m

3. Menentukan kolom kunci diantara kolom-kolom variabel yang ada, yaitu kolom yang mengandung nilai $(c_j - z_j) \geq 0$ terbesar untuk kasus maksimasi dan atau mengandung nilai $(c_j - z_j) \leq 0$ terkecil untuk kasus minimasi.
4. Tentukan baris kunci di antara baris-baris variabel, yaitu baris yang memiliki perbandingan besaran dengan nilai positif terkecil

   Rasio kuantitas ke-$i = \frac{b_i}{unsur\ kolom\ kunci\ positif}$

5. Membentuk tabel selanjutnya yaitu mengganti variabel dasar dengan memasukkan (*entering variable*) dan menghapus variabel non basis (*leaving*

variable) dari kolom tersebut, serta mentransformasikan baris variabel menggunakan rumus transformasi:

a) Baris baru selain baris kunci = baris lama – (rasio kunci x baris kunci lama), dengan rasio kunci $= \frac{unsur\ kolom\ kunci}{angka\ kunci}$

b) Baris kunci baru $= \frac{semua\ unsur\ baris\ kunci\ lama}{angka\ kunci}$

Persimpangan antara deretan kunci dan kolom adalah angka kunci.

6. Melakukan uji optimalisasi. Jika semua koefisien pada baris $(c_j - z_j)$ sudah tidak ada lagi yang bernilai positif (untuk kasus maksimasi) atau sudah tidak ada lagi yang bernilai negatif (untuk kasus minimasi), berarti tabel sudah optimal. Akan tetapi, apabila kriteria ini belum terpenuhi, maka proses diulang kembali ke langkah 3 sampai nilai semua koefisien pada baris $(c_j - z_j)$ bernilai negatif untuk kasus maksimasi atau positif untuk kasus minimasi (Dewi, Trastawati & Kartikasari, 2014).

Dalam model pemrograman linier, kendala dalam fungsi kendala dapat dibedakan berdasarkan tanda hubungan matematis yang digunakan, yaitu ≤ (kurang dari atau sama dengan), = (sama dengan), dan ≥ (lebih dari atau sama dengan). Oleh karena itu, setiap kendala dalam fungsi kendala model pemrograman linier harus ditambahkan variabel tambahan di sisi kiri menggunakan metode simpleks. Variabel tambahan ini terdiri dari variabel slack, variabel surplus, dan variabel artificial, yang membantu dalam proses pemecahan menggunakan metode simpleks (Meliana, 2019).

B. Program *Integer*

Pemrograman bilangan bulat (program *integer*) merupakan variasi dari pemrograman linier di mana solusi optimal yang dicari harus menghasilkan nilai variabel yang berupa bilangan bulat, bukan pecahan.

Dalam pemrograman bilangan bulat, terdapat syarat tambahan bahwa beberapa atau semua variabel harus memiliki nilai bilangan bulat positif. Namun, untuk mendapatkan solusi bilangan bulat dari masalah pemrograman linier, seringkali cukup menggunakan metode simpleks biasa dan kemudian membulatkan nilai pecahan dari variabel dasar. Dalam hal ini, penting untuk memastikan bahwa solusi yang diperoleh masih memenuhi semua kendala yang ada dan tidak terlalu jauh dari solusi bilangan bulat yang diinginkan. Oleh karena itu, untuk mencapai solusi bilangan bulat yang optimal, diperlukan prosedur yang rumit dan lebih lanjut.

1. Model Pemrograman *Integer*

   Ada tiga jenis model dalam pemrograman bilangan bulat (*integer*):

   a. Pemrograman Bilangan Bulat Murni, di mana semua variabel keputusan harus menjadi bilangan bulat non-negatif.
   b. Pemrograman Bilangan Bulat Campuran, di mana beberapa, tetapi tidak semua, variabel keputusan harus menjadi bilangan bulat non-negatif.
   c. Program Bilangan Bulat Biner (*Zero One Integer Programming*), di mana semua variabel keputusan harus bernilai 0 atau 1 (Purba dan Ahyaningsing, 2020).

   Bentuk umum program *integer* dapat dirumuskan sebagai berikut:

   Maksimumkan:

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

   Dengan kendala:

$$\sum_{j=1}^{n} a_{ij} x_j \ (\geq, =, \leq)\ b_i; i = 1,2, \dots, m$$

$x_j \geq 0$ merupakan bilangan cacah, $j = 1,2, \dots, n$ dimana a_{ij}, b_i, dan c_j adalah konstanta (Angeline, Iryanto & Tarigan, 2014).

2. Solusi Bulat (*Integer*)

Untuk mencapai pendekatan bulat yang optimal, sejumlah teknik dapat digunakan antara lain pembulatan, pendekatan grafik, pendekatan *Gromory (Cutting Plane Algorithm)* dan metode *Branch and Bound* (Maslihah, 2015).

3. Metode Pembulatan

Salah satu metode paling sederhana, praktis, dan efisien untuk menyelesaikan masalah pemrograman bilangan bulat adalah dengan menggunakan metode pembulatan.. Metode ini menggunakan metode simpleks standar untuk menghitung nilai variabel keputusan. Dari segi waktu, biaya dan upaya untuk mencari jawaban, metode ini layak digunakan. Metode pembulatan mungkin tidak digunakan dalam beberapa kasus, misalnya karena kemungkinan solusi pembulatan yang diperoleh kurang efektif daripada solusi akhir dan ada juga risiko kegagalan untuk mencapai hasil yang sesuai. Jika barang yang dibulatkan seperti pesawat komersial, kapal perang atau barang lain dengan nilai jual tinggi dibulatkan ke lingkaran berikutnya, hasil ini akan memiliki konsekuensi yang signifikan.

4. Metode Cutting Planes

Salah satu metode yang digunakan untuk menyelesaikan masalah pemrograman linier dengan variabel yang harus berupa bilangan bulat, termasuk bilangan bulat murni dan campuran, adalah metode *Cutting Planes*. Metode ini melibatkan penambahan kendala baru yang disebut batasan *Gomory*. Jika variabel keputusan tidak dibulatkan atau memiliki sebagian kecil dari nilai non-bulat, batasan *Gomory* diterapkan untuk memastikan solusi hanya menghasilkan bilangan bulat yang layak.

Langkah-langkah dalam metode *Cutting Planes* adalah sebagai berikut:

a. Gunakan metode simpleks untuk menyelesaikan masalah pemrograman bilangan bulat.
b. Jika solusi optimal dari variabel keputusan adalah bilangan bulat, maka proses dihentikan dan solusi dianggap sudah optimal. Jika variabel keputusan masih berupa pecahan, lanjut ke langkah berikutnya.
c. Terapkan kendala atau batasan *Gomory* dan gunakan metode dual simpleks untuk menyelesaikannya.

d. Kembali ke langkah b dan ulangi proses hingga diperoleh solusi yang optimal (Maslihah, 2015).

5. Metode *Branch and Bound*

Pada tahun 1960, A.H. Land dan A.G. Doig memperkenalkan metode *Branch and Bound*. Metode ini digunakan untuk mencari solusi optimal dalam pemrograman linier dengan variabel keputusan yang harus berupa bilangan bulat. Metode ini bekerja dengan cara menyederhanakan solusi optimal yang menghasilkan pecahan, melalui pembentukan cabang atas dan cabang bawah untuk setiap variabel keputusan yang memiliki nilai pecahan. Dengan melakukan cabang ini, setiap solusi akan menghasilkan cabang baru yang dieksplorasi untuk mencari solusi optimal dengan nilai bilangan bulat. Dalam proses ini, metode *Branch and Bound* berupaya mencapai solusi yang optimal dengan mempertimbangkan semua kemungkinan solusi yang memenuhi kriteria bilangan bulat (Purba dan Ahyaningsing, 2020).

Metode Branch and Bound adalah salah satu pendekatan yang dapat digunakan dalam menyelesaikan masalah pemrograman linier bilangan bulat. Metode ini memecah masalah menjadi sub-masalah yang mengarah pada solusi dengan membangun pohon keputusan dan melakukan pemangkasan (*bounding*) untuk mencapai solusi optimal. Dalam perbandingan dengan metode simpleks, metode Branch and Bound memiliki keunggulan dalam menyelesaikan masalah optimasi di mana nilai variabel optimal harus berupa bilangan bulat, sementara metode simpleks dapat menghasilkan nilai variabel optimal yang bukan bilangan bulat. Dengan menggunakan metode *Branch and Bound*, solusi dengan nilai bilangan bulat dapat dijelajahi secara sistematis untuk mencari solusi optimal yang memenuhi kriteria tersebut (Suryawan, Tastrawati & Kartikasari, 2016)

Metode lain dapat digunakan dalam kasus *integer programming* seperti algoritma cutting plane atau biasa dikenal dengan *cutting plane algorithm*. Metode *Branch and Bound* dalam hasil penyelesaian berbeda dengan metode cutting plane yaitu menggunakan nilai pecahan yang lebih optimal. Metode *Cutting Plane* terbukti tidak efektif dalam memecahkan masalah pemrograman bilangan bulat terlepas dari ukurannya. Pada metode *Cutting Plane* sering terdapat variasi batasan urutan yang membuat pemecahan masalah yang sangat kompleks menjadi relatif sederhana dari sudut pandang komputer (Suryawan, dkk, 2016).

Langkah-langkah berikut diikuti ketika menggunakan metode *Branch and Bound*:

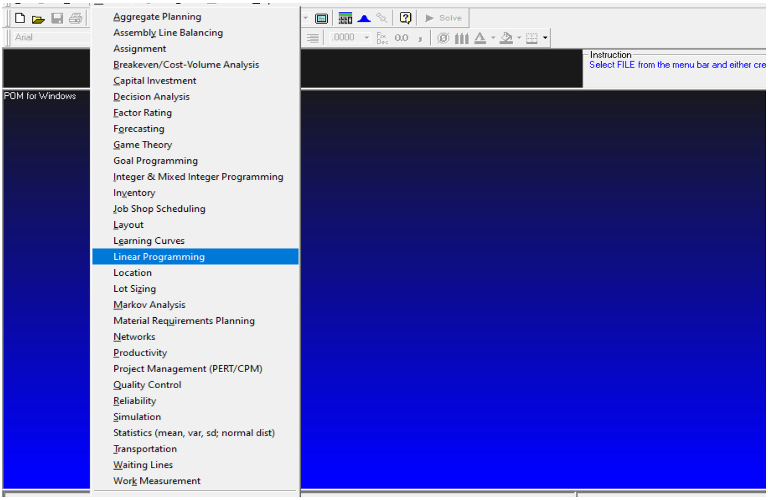
a. Gunakan metode simpleks untuk menyelesaikan model pemrograman linier.

b. Periksa solusi optimal, jika variabel dasar yang diharapkan bernilai bulat, maka solusi optimal telah tercapai. Namun, jika ada variabel dasar yang tidak bernilai bulat, lanjutkan ke langkah berikutnya.
c. Pilih variabel dengan pecahan terbesar dan buat sub-masalah baru dengan menggantinya dengan bilangan bulat untuk semua variabel yang terlibat dalam sub-masalah tersebut.
d. Kemudian untuk basis X_j^* variabel yang nilainya real dirubah menjadi integer X_j yang batasnya $X_j^* \leq X_j \leq X_j^* + 1$, tetapi karena range tersebut tidak memberikan penyelesaian *integer*, maka konsekuensinya nilai *integer* X_j harus memenuhi salah satu syarat yaitu: $X_j \geq X_j^*$ atau $X_j \leq X_j^* + 1$.
e. Memecahkan model pemrograman linier dengan menambahkan kendala baru ke setiap sub-masalah. Kembali ke langkah d jika solusi yang dimaksud sudah *integer*. Kembali ke langkah c jika angkanya bukan bilangan bulat.
f. Jika salah satu sub-masalah memiliki solusi bulat dan sub-masalah lainnya tidak memiliki solusi (tidak mungkin tercapai), maka tidak ada cabang baru yang dibuat dan proses dihentikan.

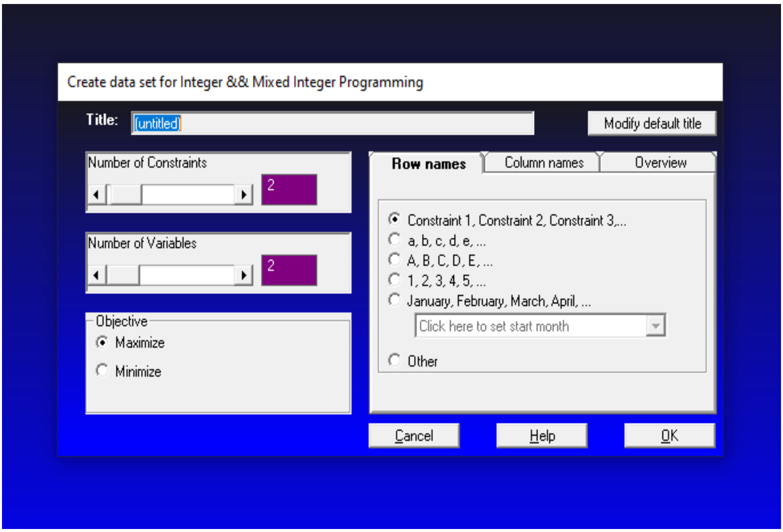
g. Pilih solusi optimal, jika ada beberapa sub-masalah dengan solusi bulat. Untuk fungsi tujuan maksimum, pilih solusi dengan nilai Z tertinggi, sedangkan untuk fungsi tujuan minimum, pilih solusi dengan nilai Z terendah sebagai solusi optimal (Safitri, Basriati & Najmi, 2020)

C. Program POM Sofware Windows

Program Production and Operation Management (POM) adalah program komputer yang menangani masalah manufaktur dan operasi yang bersifat matematis. Aplikasi keputusan misalnya untuk menentukan kombinasi pembuatan yang tepat guna mendapatkan keuntungan yang maksimal, dibuat POM for Windows dengan tampilan grafis yang menarik dan kemudahan pengoperasian. Untuk meminimalkan biaya perawatan, pesanan pembelian barang harus dibuat sedemikian rupa untuk menentukan di mana karyawan ditugaskan untuk mencapai hasil yang maksimal.



Gambar 2. 1 Tampilan Layar utama program POM for Windows



Gambar 2. 2 Tampilan di POM for Windows setelah mengklik *Integer Programming*

Layar utama POM for Windows pada dasarnya terdiri dari:

a. Title Bar: Program Name, The Control Main Box dan beberapa button untuk layar POM seperti Maxinize, Minimize dan Close
b. Menu Bar: File, View, Help ,Module, Tables, Tools, Window dan Edit
c. Button Bar atau Tool Bar: Command Bar (isinya tergantung program dan modul), Status Panel, Instrustion Panel, Data Table, Extra Data Area, Annotation Area.

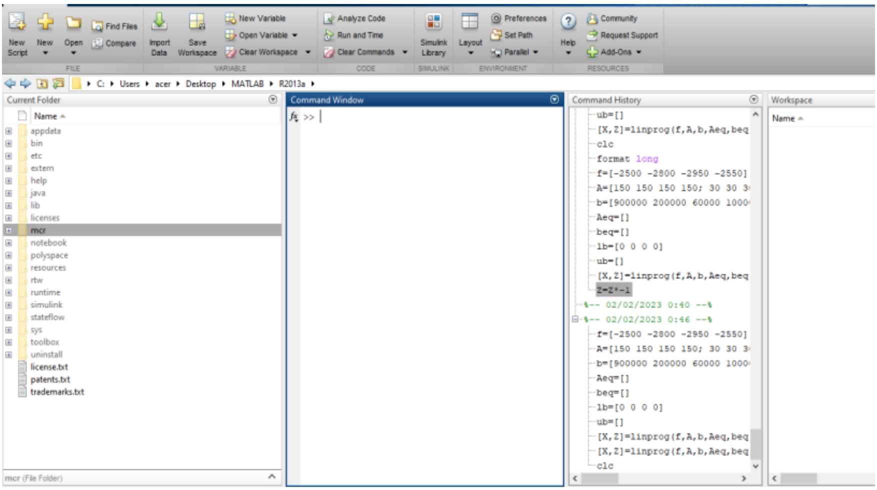
D. Penyelesaian dengan Menggunakan MATLAB

Dalam mengatasi tantangan pengoptimalan, sangat penting untuk menggunakan perangkat lunak. Terutama ketika mencoba mencari solusi optimal untuk suatu masalah, dengan melakukan banyak iterasi. Salah satu program yang banyak digunakan untuk menyelesaikan masalah optomasi adalah MATLAB.

MATLAB adalah istilah untuk *Matrix* Laboratory. MATLAB adalah bahasa pemrograman yang dimaksudkan untuk digunakan dalam pemrosesan teknis. Sebagian besar masalah yang harus diselesaikan dengan menggunakan matriks dan vektor. Ada banyak alat di

MATLAB. *Toolbox* adalah pustaka fungsionalitas MATLAB yang digunakan untuk memecahkan masalah tertentu.

Ada berbagai jenis sistem yabg dapat digunakan dengan MATLAB, dan tugas akhir ini didasarkan pada sistem windows. Untuk membuka MATLAB yaitu dengan mengklik dua kali pintasan di keyboard komputer, klik mulai, pilih semua program, lalu arahkan penunjuknya ke MATLAB dan klik. Kemudian akan ditunjukkan tampilan seperti pada Gambar 2.3 berikut:



Gambar 2. 3 Dekstop MATLAB

Berikut adalah beberapa bagian dari desktop MATLAB

1. *Command Window*

   Window ini adalah jendela utama MATLAB. Ini adalah tempat untuk memanggil fungsi, mendeklarasikan variabel, menjalankan proses, dan melihat isinya.

2. *Command History*

   Window ini berfungsi untuk menyimpan perintah-perintah apa saja yang sebelumnya dilakukan oleh pengguna terhadap MATLAB.

3. *Current Directory*

   ketika menggunakan MATLAB, isi *directory* kerja ditampilkan di jendela ini. Bergantung pada *directory* kerja yang dimintai, *directory* ini dapat diubah. Di folder kerja tempat file program MATLAB disimpan, ada alamat *directory default* kerja yang diinginkan.

4. *Workspace*

   Fungsi ruang kerja akan menampilkan semua variabel yang digunakan di MATLAB. Pengguna dapat melihat isi semua data dengan mengklik dua kali pada variabel jika itu adalah matriks data yang besar. *Window array editor* yang berisi informasi untuk setiap variabel yang dipilih oleh pengguna akan secara otomatis ditampilkan di MATLAB.

   MATLAB akan memberi fungsi *help* dan berisi dokumentasi tentang MATLAB. Pengguna dapat mengklik tombol pada *toolbar* untuk memulai fungsi ini, atau menulis *helpwin* pada *command window. Fitur demonstrasi* MATLAB, yang berisi tutorial video dan contoh program yang dapat digunakan dengan MATLAB juga tersedia (Ulfiana, 2009).

E. Kajian Pustaka

Berdasarkan hasil penelitian terdahulu yang berkaitan dengan metode *Branch and Bound,* terdapat hasil penelitian yang dapat dijadikan dasar acuan, antara lain:

1. Supatimah, Sri Siti, dkk (2019) yang berjudul "Optimasi Keuntungan dengan Metode *Branch and Bound*". Berdasarkan pada penelitian tersebut dapat disimpulkan bahwa hasil perhitungan keuntungan optimum di *Sentral Me Laundry* adalah dengan mencuci Badcover sebanyak 53 kg, Boneka sebanyak 188 kg, Pakaian sebanyak 1350 kg, Selimut sebanyak 101 kg dan memperoleh keuntungan sebesar Rp 5.095.420.
2. Syaddad, Achmad (2021) yang berjudul Analisis Metode *Branch and Bound* pada Optimalisasi Laba Industri Tambang Batu Bata Putih. Berdasarkan pada penelitian yang dilakukan menyatakan bahwa untuk mendapatkan laba yang optimal, perusahaan memproduksi batu bata putih sebanyak 42 buah, batu bata "Bhator", 4681 buah batu bata "Bhetah", dan 76 buah batu bata "Sendhi" sehingga menghasilkan keuntungan sebesar Rp 856.515 per hari.
3. Meliana, dkk (2019) yang berjudul "Penerapan Algoritma *Branch and Bound* dalam Menentukan Optimasi Jumlah Produksi Roti (Studi Kasus: CV Sedap Sari Bakery)". Berdasarkan pada penelitian yang dilakukan bahwa

penerapan algoritma *branch and bound* dalam mencari keuntungan optimal produksi roti perhari menghasilkan keuntungan sebesar Rp 5.101.100. Jumlah kombinasi produk roti yaitu roti bolu sebanyak 423 pcs, roti gulung sebanyak 166 bungkus, roti mandarin sebanyak 100 bungkus, roti tiga rasa sebanyak 175 bungkus, roti coklat sebanyak 423 bungkus dan roti isi coklat 342 bungkus.

4. Purba, Sari Devi dan Faiz Ahyaningsih (2020) yang berjudul "Integer Programming Dengan Menggunakan Metode Branch and Bound Dalam Optimasi Jumlah Produksi Setiap Jenis Roti Pada PT. Arma Anugerah Abadi". Berdasarkan pada penelitian yang dilakukan bahwa untuk mendapatkan keuntungan yang optimal, perusahaan memproduksi roti coklat sebanyak 200 bungkus, roti coklat keju sebanyak 850 bungkus, roti kelapa sebanyak 250 bungkus, roti kacang merah sebanyak 500 bungkus dan roti srikaya sebanyak 600 bungkus dengan keuntungan sebesar Rp 32.850.000.
5. Safitri, Elfira, dkk (2020) yang berjudul "Penerapan Metode Branch and Bound dalam Optimalisasi Produk Mebel (Studi Kasus: Toko Mebel di Jalan Marsan Panam)". Berdasarkan pada penelitian tersebut bahwa mebel *furniture* di Jalan Marsan Panam memperoleh keuntungan produksi sebesar Rp 14.250.000 dengan menerima pesanan produk mebel jenis tempat tidur sebanyak 4 unit,

lemari 3 pintu sebanyak 4 unit, lemari 2 pintu sebanyak 2 unit dan meja makan sebanyak 3 unit dan mebel *furniture* di Jalan Marsan Panam tidak menerima pesanan untuk jenis produk sofa, kursi tamu, bofet tv, meja rias dan pelaminan.

BAB III
METODE PENELITIAN

A. Pendekatan Penelitian

Metode ini didasarkan pada rumusan masalah dan tujuan penelitian yang telah ditetapkan pada bab awal. Penelitian ini memulai dari realitas yang jelas dan teramati, menggunakan pendekatan deduktif. Semua jenis penelitian ini dilakukan secara langsung di lapangan.

B. Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang berasal dari bahan baku pembuatan roti di Safa Bakery, yang terletak di Kecamatan Kayen, Kabupaten Pati. Penelitian ini mencakup pengumpulan informasi mengenai persediaan bahan baku, harga jual, dan biaya produksi untuk empat jenis roti yaitu Roti Kasur, Roti Kepang, Roti Bantal, dan Roti Sobek.

C. Analisis Data

Penelitian disusun dengan langkah-langkah sebagai berikut:

1. Studi Lapangan
   a. Pengamatan (Observasi) adalah suatu cara mendapatkan informasi secara langsung dengan peninjauan ke Safa Bakery.

b. Wawancara (*Interview*) yaitu dilakukan secara langsung dengan pemilik dan karyawan Safa Bakery dengan mengajukan pertanyaan dan menjawabnya.

2. Studi Pustaka

Studi pustaka terdiri dari pengumpulan dan pemeriksaan berbagai informasi yang berkaitan dengan buku atau jurnal, serta makalah tentang masalah yang akan ditangani. Metode *Branch and Bound* merupakan teori dasar yang diterapkan untuk mencari solusi dari permasalahan yang ada pada usaha roti.

3. Pengumpulan Data

Jumlah produk roti, bahan baku yang diproduksi, biaya produksi, harga jual per jenis roti dan keuntungan pada bulan Februari 2022 adalah semua data yang dikumpulkan dari sebuah bisnis perdagangan.

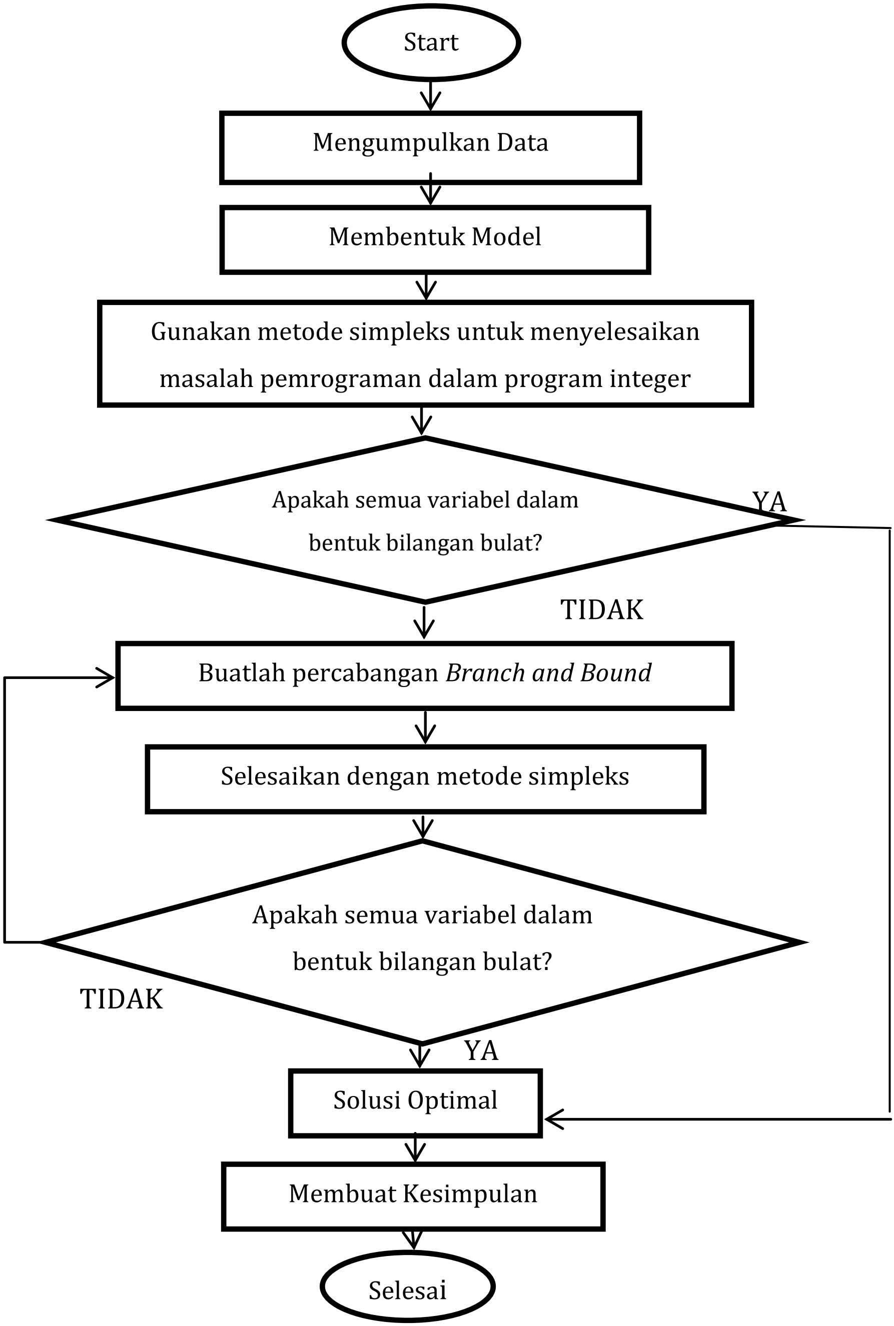
4. Pengolahan Data

Pemrosesan data dilakukan sesuai dengan langkah-langkah berikut:

a. Dalam bentuk program linier, modelkan fungsi tujuan dan fungsi kendala yang ada. Fungsi tujuan merupakan tujuan yang ingin dicapai, misalnya maksimalkan keuntungan. Fungsi kendala adalah batasan-batasan yang harus dipenuhi, seperti persediaan bahan baku, kapasitas produksi, harga jual minimum, dan biaya produksi maksimum.

b. Untuk menemukan jumlah optimal produk yang dapat diproduksi untuk keuntungan maksimal, gunakan metode simpleks dengan program POM for Windows dan MATLAB untuk menghitung nilai variabel menggunakan program ini.
c. Dengan menggunakan metode *Branch and Bound*, dapat mengubah jumlah produk yang bukan bilangan bulat yang didapatkan dari metode simpleks menjadi bentuk bilangan bulat.
d. Kesimpulan dari hasil perhitungan yang disajikan sebagai saran yang dapat digunakan dalam usaha perdagangan diambil setelah diperoleh jumlah optimal produk tipe bilangan bulat.

Alur penelitian ini telah diatur dalam kerangka *flowchart* sehingga dapat dianalisis dengan lebih mudah sebagai berikut:



Gambar 3.1. *Flowchart* Metode *Branch and Bound*

BAB IV

HASIL DAN PEMBAHASAN

Pada bagian ini, akan dibahas mengenai pengoptimalan proses produksi roti di Safa Bakery menggunakan metode Branch and Bound. Data yang digunakan dari Safa Bakery meliputi:

a. Data persediaan bahan baku untuk keempat jenis roti serta data komposisi bahan baku yang dibutuhkan untuk membuat satu buah roti.

b. Data biaya produksi, harga jual dan keuntungan penjualan dari keempat jenis roti.

Berikut tabel yang berisi data komposisi bahan baku:

Tabel 4.1. Data persediaan bahan baku dari 4 jenis roti dan data komposisi bahan baku untuk 1 buah roti

Bahan Baku Roti	Jenis Roti				Persediaan Maksimal	Satuan
	Roti Kasur	Roti Kepang	Roti Bantal	Roti Sobek		
Terigu	150	150	150	150	900000	Gram
Gula	30	30	30	30	200000	Gram
Telur	10	10	10	10	60000	Gram
Margarin	13	13	20	13	100000	Gram
Garam	10	10	10	10	55000	Gram
Susu Bubuk	40	40	40	40	220000	Gram
Ragi	10	10	10	10	60000	Gram
Coklat	56	0	35	0	200000	Gram
Pisang	0	0	60	0	90000	Gram
Selai	0	0	0	97	170000	Gram
Seres	0	60	0	0	58000	Gram

Tabel 4.2. Data Biaya Produksi, Harga Jual dan Keuntungan Penjualan Roti

No	Jenis Roti	Harga Penjualan	Biaya Produksi	Keuntungan (Rupiah)
1.	Roti Kasur	Rp 10.000	Rp 7.500	Rp 2.500
2.	Roti Kepang	Rp 10.000	Rp 7.200	Rp 2.800
3.	Roti Bantal	Rp 13.000	Rp 10.050	Rp 2.950
4.	Roti Sobek	Rp 11.000	Rp 8.450	Rp 2.550

Untuk mengetahui permasalahan optimalisasi hasil produksi pada Safa Bakery ke dalam bentuk program linier, maka dilakukan langkah-langkah sebagai berikut:

A. Memodelkan Masalah Optimalisasi Produksi Roti ke dalam Bentuk Program Linier

Perancangan model program linier adalah langkah paling penting dalam program linier. Permintaan untuk keterangan mengenai tujuan dan batasan yang membatasinya tercakup dalam langkah ini. Beberapa elemen yang digunakan untuk pengembangan model pemrograman linier, seperti perumusan variabel keputusan, fungsi pembatas kendala, fungsi tujuan dan variabel pembatas, akan diterapkan untuk mengembangkan model dari perumusan masalah saat ini (Rafflesia & Widodo, 2014)

1. Variabel Keputusan

   Identifikasi variabel-variabel yang mempengaruhinya merupakan langkah awal dalam memodelkan suatu masalah. Berdasarkan Tabel 4.1 terdapat empat variabel keputusan yang menunjukkan jumlah roti kasur, roti kepang, roti bantal dan roti sobek yang diproduksi periode bulan Februari di Safa Bakery. Variabel keputusan dalam penelitian ini adalah:

   x_1 = Banyak Roti Kasur yang diproduksi

   x_2 = Banyak Roti Kepang yang diproduksi

   x_3 = Banyak Roti Bantal yang diproduksi

   x_4 = Banyak Roti Sobek yang diproduksi

2. Fungsi Tujuan

   Memaksimalkan keuntungan dari semua produksi roti adalah tujuan dari kegiatan ini. Keuntungan dari setiap jenis produksi roti dihitung dengan menggunakan koefisien untuk setiap variabel keputusan. Fungsi tujuan untuk produksi roti pada Safa Bakery dapat dilihat pada tabel 4.1 yaitu:

   Maksimalkan:

   $$Z = 2500x_1 + 2800x_2 + 2950x_3 + 2550x_4 \qquad (4.1)$$

3. Pembentukan Fungsi Kendala

   Fungsi kendala adalah batasan-batasan yang harus diperhatikan dalam suatu masalah, yang juga merupakan hubungan linier dengan variabel keputusan. Dalam konteks produksi, kendala ini sering disebut sebagai hambatan atau batasan produksi. Secara sederhana, fungsi kendala adalah fungsi yang memuat batasan-batasan tertentu yang harus dipatuhi dalam mencapai solusi optimal. Dalam hal ini fungsi yang merupakan bentuk dan keterbatasan dalam produksi roti adalah sebagai berikut:

Terigu : $150x_1 + 150x_2 + 150x_3 + 150x_4 \leq 900000$ (4.2)

Gula : $30x_1 + 30x_2 + 30x_3 + 30x_4 \leq 200000$ (4.3)

Telur : $10x_1 + 10x_2 + 10x_3 + 10x_4 \leq 60000$ (4.4)

Margarin: $13x_1 + 13x_2 + 20x_3 + 13x_4 \leq 100000$ (4.5)

Garam : $10x_1 + 10x_2 + 10x_3 + 10x_4 \leq 55000$ (4.6)

Susu Bubuk: $40x_1 + 40x_2 + 40x_3 + 40x_4 \leq 220000$ (4.7)

Ragi: $10x_1 + 10x_2 + 10x_3 + 10x_4 \leq 60000$ (4.8)

Cokelat: $56x_1 + 35x_3 \leq 200000$ (4.9)

Pisang : $60x_3 \leq 90000$ (4.10)

Selai: $97x_4 \leq 170000$ (4.11)

Seres: $60x_2 \leq 58000$ (4.12)

dengan kendala variabel:

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

4. Memodelkan Masalah Optimasi dalam Bentuk Pemrograman Linier

Bentuk umum pemrograman linier dalam masalah optimasi hasil produksi roti adalah sebagai berikut:

Maksimalkan $Z = 2500x_1 + 2800x_2 + 2950x_3 + 2550x_4$

Dengan kendala:

Terigu : $150x_1 + 150x_2 + 150x_3 + 150x_4 \leq 900000$

Gula : $30x_1 + 30x_2 + 30x_3 + 30x_4 \leq 200000$

Telur : $10x_1 + 10x_2 + 10x_3 + 10x_4 \leq 60000$

Margarin: $13x_1 + 13x_2 + 20x_3 + 13x_4 \leq 100000$

Garam : $10x_1 + 10x_2 + 10x_3 + 10x_4 \leq 55000$

Susu Bubuk: $40x_1 + 40x_2 + 40x_3 + 40x_4 \leq 220000$

Ragi : $10x_1 + 10x_2 + 10x_3 + 10x_4 \leq 60000$

Cokelat: $56x_1 + 35x_3 \leq 200000$

Pisang : $60x_3 \leq 90000$

Selai : $97x_4 \leq 170000$

Seres : $60x_2 \leq 58000$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

B. Menyelesaikan Model Matematika dengan Menggunakan Metode Simpleks

Hal pertama yang harus dilakukan adalah mengubah rumus pemrograman linier ke bentuk strandar metode simpleks. Langkah-langkah metode simpleks pada masalah optimalisasi produksi roti pada Safa Bakery adalah sebagai berikut:

1. Mengubah fungsi tujuan dan fungsi kendala ke dalam bentuk standar

Maksimumkan:

Fungsi tujuan: $Z - 2500x_1 - 2800x_2 - 2950x_3 - 2550x_4 = 0$

Fungsi kendala:

$150x_1 + 150x_2 + 150x_3 + 150x_4 + \ S_1 = 900000$

$30x_1 + 30x_2 + 30x_3 + 30x_4 + S_2 = 200000$

$10x_1 + 10x_2 + 10x_3 + 10x_4 + S_3 = 60000$

$13x_1 + 13x_2 + 20x_3 + 13x_4 + S_4 = 100000$

$10x_1 + 10x_2 + 10x_3 + 10x_4 + S_5 = 55000$

$40x_1 + 40x_2 + 40x_3 + 40x_4 + S_6 = 220000$

$10x_1 + 10x_2 + 10x_3 + 10x_4 + S_7 = 60000$

$56x_1 + 35x_3 + S_8 = 200000$

$60x_3 + S_9 = 90000$

$97x_4 + S_{10} = 170000$

$60x_2 + S_{11} = 58000$

$x_1, x_2, x_3, x_4, S_1, S_2, S_3 S_4, S_5, S_6, S_7, S_8, S_9, S_{10} S_{11}, \geq 0$

dimana S_1 sampai S_{11} merupakan variabel *slack*. Variabel *slack* adalah variabel tambahan yang mewakili persediaan bahan baku yang tidak digunakan.

2. Tempatkan persamaan-persamaan ke dalam tabel 4.3 berikut:

Tabel 4.3. Tabel Awal Simpleks

Variabel Dasar	Z	x_1	x_2	x_3	x_4	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	Nilai Kunci (NK)
Z	1	-2500	-2800	-2950	-2550	0	0	0	0	0	0	0	0	0	0	0	0
S_1	0	150	150	150	150	1	0	0	0	0	0	0	0	0	0	0	900000
S_2	0	30	30	30	30	0	1	0	0	0	0	0	0	0	0	0	200000
S_3	0	10	10	10	10	0	0	1	0	0	0	0	0	0	0	0	60000
S_4	0	13	13	20	13	0	0	0	1	0	0	0	0	0	0	0	100000
S_5	0	10	10	10	10	0	0	0	0	1	0	0	0	0	0	0	55000
S_6	0	40	40	40	40	0	0	0	0	0	1	0	0	0	0	0	220000
S_7	0	10	10	10	10	0	0	0	0	0	0	1	0	0	0	0	60000
S_8	0	56	0	35	0	0	0	0	0	0	0	0	1	0	0	0	200000
S_9	0	0	0	60	0	0	0	0	0	0	0	0	0	1	0	0	90000
S_{10}	0	0	0	0	97	0	0	0	0	0	0	0	0	0	1	0	170000
S_{11}	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	1	58000

= Warna kuning menunjukkan kolom kunci
= Warna merah menunjukkan baris kunci
= Warna hijau menunjukkan kolom kunci

3. Tentukan kolom kunci

Kolom kunci adalah kolom yang dipilih sebagai basis untuk melakukan penggantian dalam tabel simpleks. Untuk menentukan kolom kunci, perlu mencari kolom dengan nilai koefisien angka terbesar yang bernilai negatif dalam fungsi tujuan. Dalam kasus ini, terpilih kolom x_3 dengan nilai koefisien -2950 sebagai kolom kunci.

x_3
-2950
150
30
20
10
40
10
35
60
0
0

4. Pilihlah baris kunci

Baris yang menjadi dasar untuk melakukan perubahan pada tabel simpleks disebut baris kunci. Untuk menentukan baris kunci, dapat dilakukan dengan mencari indeks dari setiap baris dengan membagi nilai pada kolom NK (nilai kunci) dengan nilai pada kolom kunci.

S_9	0	0	0	60	0	0	0	0	0	0	0	0	0	1	0	0	90000

5. Tetapkan nilai untuk baris kunci yang akan diubah

Seluruh nilai baris kunci diubah dengan cara membagi seluruh nilai-nilai baris kunci dengan angka kunci. Angka kunci diperoleh dari angka yang diapit oleh kolom kunci dan baris kunci.

Baris kunci

Nilai	(60)	0	0	0	60	0	0	0	0	0	0	0	0	0	1	0	0	90000	(÷)
Baru	=	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	

6. Mengganti semua nilai-nilai selain pada baris kunci

Baris Baru = baris lama - (koefisien kolom kunci X baris kunci baru)

Contoh:

Baris S_1:

Baris lama

S_1	0	150	150	150	150	1	0	0	0	0	0	0	0	0	0	0	90000

Koefisien kolom kunci x baris kunci baru:

Koefisien kolom kunci

x_3
-2950
150
30
20
10
40
10
35
60

0
0

Dikali

Baris kunci baru

S_9	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1.500

ITERASI 1

Nilai Baru (NB):

Baris S_1

		0	150	150	150	150	1	0	0	0	0	0	0	0	0	0	0	900000	
	150	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	150	0	0	0	0	0	0	0	0	0	2,5	0	0	225000	(-)
NB	S_1	0	150	150	0	150	1	0	0	0	0	0	0	0	-2,5	0	0	675000	

Baris S_2

		0	30	30	30	30	0	1	0	0	0	0	0	0	0	0	0	200000	
	30	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	30	0	0	0	0	0	0	0	0	0	0,51	0	0	45000	(-)
NB	S_2	0	30	30	0	30	0	1	0	0	0	0	0	0	-0,51	0	0	155000	

Baris S_3

		0	10	10	10	10	0	0	1	0	0	0	0	0	0	0	0	60000	
	30	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	10	0	0	0	0	0	0	0	0	0	0,17	0	0	15000	(-)
NB	S_3	0	10	10	0	10	0	0	1	0	0	0	0	0	-0,17	0	0	45000	

Baris S_4

		0	13	13	20	13	0	0	0	1	0	0	0	0	0	0	0	100000	
	20	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	20	0	0	0	0	0	0	0	0	0	0,34	0	0	30000	(-)
NB	S_4	0	13	13	0	13	0	0	0	1	0	0	0	0	-0,34	0	0	70000	

Baris S_5

		0	10	10	10	10	0	0	0	0	1	0	0	0	0	0	0	55000	
	10	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	10	0	0	0	0	0	0	0	0	0	0,17	0	0	15000	(-)
NB	S_5	0	10	10	0	10	0	0	0	0	1	0	0	0	-0,17	0	0	40000	

Baris S_6

		0	40	40	40	40	0	0	0	0	0	1	0	0	0	0	0	220000	
	40	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	40	0	0	0	0	0	0	0	0	0	0,68	0	0	60000	(-)
NB	S_6	0	40	40	0	40	0	0	0	0	0	1	0	0	-0,68	0	0	160000	

Baris S_7

		0	10	10	10	10	0	0	0	0	0	0	1	0	0	0	0	60000	
	10	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	10	0	0	0	0	0	0	0	0	0	0,17	0	0	15000	(-)
NB	S_7	0	10	10	0	10	0	0	0	0	0	0	1	0	-0,17	0	0	45000	

Baris S_8

		0	90	0	35	0	0	0	0	0	0	0	0	1	0	0	0	200000	
	35	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	35	0	0	0	0	0	0	0	0	0	0,59	0	0	52500	(-)
NB	S_8	0	90	0	0	0	0	0	0	0	0	0	0	1	-0,59	0	0	147500	

Baris S_{10}

		0	0	0	0	97	0	0	0	0	0	0	0	0	0	1	0	170000	
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(-)
NB	S_{10}	0	0	0	0	97	0	0	0	0	0	0	0	0	0	1	0	170000	

Baris S_{11}

		0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	1	58000	
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	

		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(-)
NB	S_{11}	0	0	60	0	0	0	0	0	0	0	0	0	0	0,5	0	1	58000	

Baris Z

		1	-2500	-2800	-2950	-2550	0	0	0	0	0	0	0	0	0	0	0	0	
	2950	0	0	0	1	0	0	0	0	0	0	0	0	0	0,017	0	0	1500	
		0	0	0	2950	0	0	0	0	0	0	0	0	0	50,15	0	0	4425000	+
NB	Z	1	-2500	-2800	0	-2550	0	0	0	0	0	0	0	0	50,15	0	0	4425000	

Tabel 4.4. Tabel Baru Metode Simpleks

VD	Z	x_1	x_2	x_3	x_4	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	NK
Z	1	-2500	-2800	-2950	-2550	0	0	0	0	0	0	0	0	50,15	0	0	4425000
S_1	0	150	150	150	150	1	0	0	0	0	0	0	0	-2,5	0	0	675000
S_2	0	30	30	30	30	0	1	0	0	0	0	0	0	-0,51	0	0	155000
S_3	0	10	10	10	10	0	0	1	0	0	0	0	0	-0,17	0	0	45000
S_4	0	13	13	20	13	0	0	0	1	0	0	0	0	-0,34	0	0	70000
S_5	0	10	10	10	10	0	0	0	0	1	0	0	0	-0,17	0	0	40000
S_6	0	40	40	40	40	0	0	0	0	0	1	0	0	-0,68	0	0	160000
S_7	0	10	10	10	10	0	0	0	0	0	0	1	0	-0,17	0	0	45000
S_8	0	56	0	35	0	0	0	0	0	0	0	0	1	-0,59	0	0	147500
S_9	0	0	0	60	0	0	0	0	0	0	0	0	0	0,017	0	0	1500
S_{10}	0	0	0	0	97	0	0	0	0	0	0	0	0	0	1	0	170000
S_{11}	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	1	58000

Karena pada baris fungsi tujuan masih terdapat nilai negatif yaitu $x_1 = -2500$, $x_2 = -2800$, $x_3 = -2950$, $x_4 = -2550$, maka iterasi berlanjut dengan perbaikan. Jika tidak ada nilai negatif pada semua nilai baris fungsi tujuan maka iterasi berhenti.

C. Penyelesaian menggunakan Bantuan POM for Windows

Berikut adalah iterasi 1-5 metode simpleks dengan menggunakan bantuan POM for Windows:

Objective
- Maximize
- Minimize

Instruction
POM for Windows has selected a column to enter the basis. You may change this

Linear Programming Results

Simplex Iteration 1

Cj	Basic Variables	2500 X1	2800 X2	2950 X3	2550 X4	0 slack 1	0 slack 2	0 slack 3	0 slack 4	0 slack 5
0	slack 1	150	150	150	150	1	0	0	0	0
0	slack 2	30	30	30	30	0	1	0	0	0
0	slack 3	10	10	10	10	0	0	1	0	0
0	slack 4	13	13	20	13	0	0	0	1	0
0	slack 5	10	10	10	10	0	0	0	0	1
0	slack 6	40	40	40	40	0	0	0	0	0
0	slack 7	10	10	10	10	0	0	0	0	0
0	slack 8	56	0	35	0	0	0	0	0	0
0	slack 9	0	0	60	0	0	0	0	0	0
0	slack 10	0	0	0	97	0	0	0	0	0
0	slack 11	0	60	0	0	0	0	0	0	0
	zj	0	0	0	0	0	0	0	0	0
	cj-zj	2500	2800	2950	2550	0	0	0	0	0

Gambar 4. 1 Iterasi I Metode Simpleks dengan menggunakan POM for Windows

Terlihat pada Gambar 4.1 yang menunjukkan langkah-langkah iterasi pertama metode simpleks yang melibatkan data komposisi bahan baku untuk 1 buah roti.

Objective
(•) Maximize
( ) Minimize

Instruction
POM for Windows has selected a column to enter the basis. You may change this

Linear Programming Results

Simplex Iteration 2

Cj	Basic Variables	2500 X1	2800 X2	2950 X3	2550 X4	0 slack 1	0 slack 2	0 slack 3	0 slack 4	0 slack 5
0	slack 1	150	150	0	150	1	0	0	0	0
0	slack 2	30	30	0	30	0	1	0	0	0
0	slack 3	10	10	0	10	0	0	1	0	0
0	slack 4	13	13	0	13	0	0	0	1	0
0	slack 5	10	10	0	10	0	0	0	0	1
0	slack 6	40	40	0	40	0	0	0	0	0
0	slack 7	10	10	0	10	0	0	0	0	0
0	slack 8	56	0	0	0	0	0	0	0	0
2950	X3	0	0	1	0	0	0	0	0	0
0	slack 10	0	0	0	97	0	0	0	0	0
0	slack 11	0	60	0	0	0	0	0	0	0
	zj	0	0	2950	0	0	0	0	0	0
	cj-zj	2500	2800	0	2550	0	0	0	0	0

Gambar 4. 2 Iterasi II Metode Simpleks dengan bantuan POM for Windows

Dapat dilihat pada Gambar 4.2 bahwa pada iterasi II menunjukkan *slack* 9 keluar dari *basic variables* dan x_3 masuk ke dalam *basic variables.*

Objective
(•) Maximize
( ) Minimize

Instruction
POM for Windows has selected a column to enter the basis. You may change this b

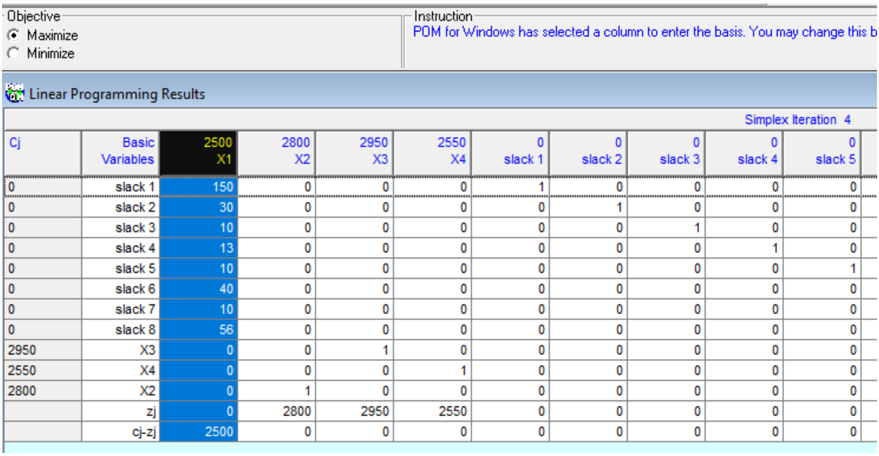
Linear Programming Results

Simplex Iteration 3

Cj	Basic Variables	2500 X1	2800 X2	2950 X3	2550 X4	0 slack 1	0 slack 2	0 slack 3	0 slack 4	0 slack 5
0	slack 1	150	0	0	150	1	0	0	0	0
0	slack 2	30	0	0	30	0	1	0	0	0
0	slack 3	10	0	0	10	0	0	1	0	0
0	slack 4	13	0	0	13	0	0	0	1	0
0	slack 5	10	0	0	10	0	0	0	0	1
0	slack 6	40	0	0	40	0	0	0	0	0
0	slack 7	10	0	0	10	0	0	0	0	0
0	slack 8	56	0	0	0	0	0	0	0	0
2950	X3	0	0	1	0	0	0	0	0	0
0	slack 10	0	0	0	97	0	0	0	0	0
2800	X2	0	1	0	0	0	0	0	0	0
	zj	0	2800	2950	0	0	0	0	0	0
	cj-zj	2500	0	0	2550	0	0	0	0	0

Gambar 4. 3 Iterasi III Metode Simpleks dengan bantuan POM for Windows

Dapat dilihat pada Gambar 4.3 bahwa Pada iterasi III menunjukkan *slack* 11 keluar dari *basic variables* dan x_2 masuk kedalam *basic variables.*



Cj	Basic Variables	2500 X1	2800 X2	2950 X3	2550 X4	0 slack 1	0 slack 2	0 slack 3	0 slack 4	0 slack 5
0	slack 1	150	0	0	0	1	0	0	0	0
0	slack 2	30	0	0	0	0	1	0	0	0
0	slack 3	10	0	0	0	0	0	1	0	0
0	slack 4	13	0	0	0	0	0	0	1	0
0	slack 5	10	0	0	0	0	0	0	0	1
0	slack 6	40	0	0	0	0	0	0	0	0
0	slack 7	10	0	0	0	0	0	0	0	0
0	slack 8	56	0	0	0	0	0	0	0	0
2950	X3	0	0	1	0	0	0	0	0	0
2550	X4	0	0	0	1	0	0	0	0	0
2800	X2	0	1	0	0	0	0	0	0	0
	zj	0	2800	2950	2550	0	0	0	0	0
	cj-zj	2500	0	0	0	0	0	0	0	0

Gambar 4. 4 Iterasi 4 Metode Simpleks dengan bantuan POM for Windows

Dapat dilihat pada Gambar 4.4 bahwa pada iterasi 4 menunjukkan *slack* 10 keluar dari *basic variables* dan x_4 masuk kedalam *basic variables.*

Objective
- (•) Maximize
- ( ) Minimize

Instruction
This is the optimal solution. Press finish to end the iterations or select a column to e

Linear Programming Results

Simplex Iteration 5

Cj	Basic Variables	2500 X1	2800 X2	2950 X3	2550 X4	0 slack 1	0 slack 2	0 slack 3	0 slack 4	0 slack 5
0	slack 1	0	0	0	0	1	0	0	0	-15
0	slack 2	0	0	0	0	0	1	0	0	-3
0	slack 3	0	0	0	0	0	0	1	0	-1
0	slack 4	0	0	0	0	0	0	0	1	-1,3
2500	X1	1	0	0	0	0	0	0	0	,1
0	slack 6	0	0	0	0	0	0	0	0	-4
0	slack 7	0	0	0	0	0	0	0	0	-1
0	slack 8	0	0	0	0	0	0	0	0	-5,6
2950	X3	0	0	1	0	0	0	0	0	0
2550	X4	0	0	0	1	0	0	0	0	0
2800	X2	0	1	0	0	0	0	0	0	0
	zj	2500	2800	2950	2550	0	0	0	0	250
	cj-zj	0	0	0	0	0	0	0	0	-250

Gambar 4. 5 Iterasi 5 Metode Simpleks dengan bantuan POM for Windows

Dapat dilihat pada Gambar 4.5 bahwa pada iterasi 5 menunjukkan *slack* 5 keluar dari *basic variables dan* x_1 masuk kedalam *basic variables.*

Objective
- (•) Maximize
- ( ) Minimize

Instruction
Iteration 5.

Linear Programming Results

(untitled) Solution

	X1	X2	X3	X4		RHS	Dual
Maximize	2500	2800	2950	2550			
Constraint 1	150	150	150	150	<=	900000	0
Constraint 2	30	30	30	30	<=	200000	0
Constraint 3	10	10	10	10	<=	60000	0
Constraint 4	13	13	20	13	<=	100000	0
Constraint 5	10	10	10	10	<=	55000	250
Constraint 6	40	40	40	40	<=	220000	0
Constraint 7	10	10	10	10	<=	60000	0
Constraint 8	56	0	35	0	<=	200000	0
Constraint 9	0	0	60	0	<=	90000	7,5
Constraint 10	0	0	0	97	<=	170000	,52
Constraint 11	0	60	0	0	<=	58000	5
Solution->	1280,76	966,67	1500	1752,58		14802630	

Gambar 4. 6 Solusi dari Hasil Iterasi dengan bantuan POM for Windows

Dari hasil iterasi dengan menggunakan POM for Windows V4, diperoleh hasil yang optimal yaitu:

$$x_1 = 1280,76$$

$$x_2 = 966,67$$

$$x_3 = 1500$$

$$x_4 = 1752,58$$

Sehingga Roti Kasur yang harus diproduksi dalam satu periode adalah 1280,76 roti, Roti Kepang sebanyak 966,67 roti, Roti Bantal sebanyak 1500 roti, dan Roti Sobek sebanyak 1752,58 roti dengan keuntungan Rp 14.802.630. Namun permasalahan ini belum valid karena solusi yang dibutuhkan adalah solusi berupa bilangan bulat maka akan digunakan metode *Branch and Bound* sehingga solusi yang dihasilkan berupa bilangan bulat.

D. Penyelesaian dengan Menggunakan MATLAB R2013a

Persamaan-persamaan model matematika selain diselesaikan dengan POM for Windows juga dapat diselesaikan dengan menggunakan bantuan perangkat lunak *Matrix Laboratory* (MATLAB) yang merupakan *software* berbasiskan fungsi-fungsi matriks. Penggunaan MATLAB R2013a ini membantu menyelesaikan permasalahan-permasalahan yang sering ditemui diberbagai bidang dengan menggunakan model matematika permasalahan program linier. Dalam memecahkan permasalahan program linier menggunakan MATLAB R2013a dapat dilakukan dengan menyelesaikan program linier pada *command window* dan dibantu oleh beberapa formula yang dapat digunakan.

Formula-formula yang digunakan untuk menyelesaikan masalah program linier di *command window* ditunjukkan pada tabel berikut:

Tabel 4.5. Formula-formula untuk menyelesaikan masalah program linier di *command window*

No.	Formula	Fungsi
1.	*F*	Vektor koefisien fungsi tujuan
2.	*A*	Matriks koefisien fungsi kendala pertidaksamaan
3.	*B*	Vektor batas fungsi kendala pertidaksamaan
4.	*Aeq*	Matriks koefisien fungsi kendala persamaan
5.	*beq*	Vektor batas fungsi kendala persamaan
6.	*lb*	Vektor batas bawah variabel keputusan
7.	*ub*	Vektor batas atas variabel keputusan

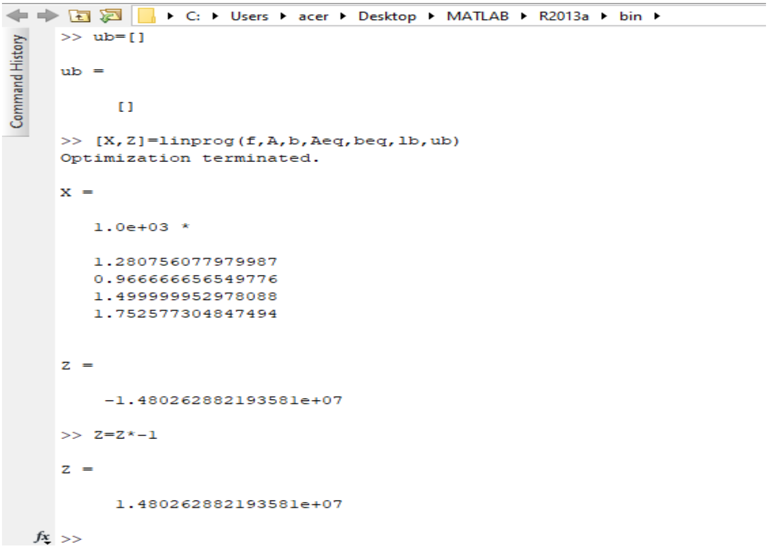
Penulisan formula-formula yang terdapat di atas dapat dituliskan di *command window* sebagai berikut:

[X,Z]= *linprog*(*f,A,b,Aeq,beq,lb,ub*)

Dimana X merupakan variabel keputusan yang akan diambil dan Z adalah hasil dari fungsi tujuan (Febrianti & Harahap, 2021). Kemudian masukkan dan mengolah data ke dalam MATLAB dari persamaan fungsi tujuan dan variabel yang diperlukan utuk menghasilkan suatu nilai optimal dari suatu titik seperti pada gambar berikut:

```
>> f=[-2500 -2800 -2950 -2550]

f =

      -2500       -2800       -2950       -2550

>> A=[150 150 150 150; 30 30 30 30;10 10 10 10;13 13 20 13;10 10 10 10;40 40 40 40;10 10 10 10;56 0 35 0;0 0 60 0;0 0 0 97;0 60 0 0]

A =

   150   150   150   150
    30    30    30    30
    10    10    10    10
    13    13    20    13
    10    10    10    10
    40    40    40    40
    10    10    10    10
    56     0    35     0
     0     0    60     0
     0     0     0    97
     0    60     0     0

>> b=[900000 200000 60000 100000 55000 220000 60000 200000 90000 170000 58000]

b =

  Columns 1 through 8

      900000      200000       60000      100000       55000      220000       60000      200000
```

```
  Columns 9 through 11

       90000      170000       58000

>> Aeq=[]

Aeq =

     []

>> beq=[]

beq =

     []

>> lb=[0 0 0 0]

lb =

     0     0     0     0

>> ub=[]

ub =

     []

>> [X,Z]=linprog(f,A,b,Aeq,beq,lb,ub)
```



Gambar 4. 7 Solusi dari Hasil Iterasi dengan bantuan POM for Windows

Program MATLAB versi R2013a tersebut, didapatkan hasil maksimal yang merupakan nilai terbaik dalam meningkatkan keuntungan yaitu dengan memproduksi roti kasur sebanyak 1.280,76 roti, roti kepang sebanyak 966,67 roti, roti bantal sebanyak 1.500 roti, dan roti sobek sebanyak 1.752,58 roti dengan keuntungan sebesar Rp 14.802.630. Hasil yang diperoleh tersebut sama dengan hasil yang diperoleh dengan menggunakan bantuan POM for windows.

E. Metode *Branch and Bound*

Langkah pertama yang dilakukan dalam pencarian solusi *integer* adalah dengan menentukan batas atas (BA) dan batas bawah (BB) dari nilai solusi yang telah diperoleh menggunakan metode *simpleks* berbantuan POM for Windows dan MATLAB. Batas atas (BA) merupakan solusi awal yang didapatkan sebelumnya yaitu $x_1 = 1280{,}76$, $x_2 = 966{,}67$, $x_3 = 1500$, $x_4 = 1752{,}58$ dengan keuntungan maksimalnya (Z) adalah Rp 14.802.630. Sedangkan batas bawah (BB) merupakan hasil pembulatan ke bawah dari solusi awal yang didapatkan sebelumnya yaitu $x_1 = 1280$, $x_2 = 966$, $x_3 = 1500$, $x_4 = 1752$ dengan keuntungan maksimalnya (Z) adalah Rp 14.797.400. Nilai keuntungan dengan pembulatan ke bawah dijadikan sebagai batas bawah (BB).

Setelah batas atas dan batas bawah ditentukan, maka selanjutnya memilih variabel keputusan untuk melakukan pencabangan (*branching*). Dipilih variabel yang mempunyai selisih pecahan terbesar dengan bilangan bulat dari masing-masing variabel yaitu x_1 sebesar 1280,76, maka x_1 dicabangkan menjadi sub-masalah 1 dan sub-masalah 2 dengan tambahan kendala untuk sub-masalah 1 adalah $x_1 \geq 1281$ dan untuk sub-masalah 2 adalah $x_1 \leq 1280$.

Sehingga diperoleh hasil sebagai berikut:

Iterasi 1:

1. Sub-masalah 1

Maksimumkan : (4.1)

Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \geq 1281$

Hasil dari metode simpleks adalah sebagai berikut:

Solusi sub-masalah 1: $x_1 = 1281$, $x_2 = 966,67$, $x_3 = 1500$, $x_4 = 1752,33$ dengan Z = 14.802.620

2. Sub-masalah 2

   Maksimumkan : (4.1)

   Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \leq 1280$

   Hasil dari metode simpleks adalah sebagai berikut:

   Solusi sub-masalah 2: $x_1 = 1280$, $x_2 = 966,67$, $x_3 = 1500$, $x_4 = 1752,58$ dengan Z = 14.800.740

Langkah selanjutnya adalah menghitung nilai solusi dalam Z dari setiap sub-problem, apakah lebih kecil atau lebih besar dari nilai batas atas. Jika nilai solusi yang diperoleh lebih besar dari batas atas, maka solusi tersebut tidak layak karena jika disubstitusikan ke dalam salah satu kendala maka akan diperoleh kesulitan yang ada. Sedangkan nilai solusi yang diperoleh lebih rendah dari batas atas, maka tidak optimal.

Karena nilai solusi dari sub-masalah 1 dan 2 tidak lebih besar dari batas atas dan tidak lebih kecil dari nilai batas bawah yaitu sub-masalah 1, BA= 14.802.620 ≤ 14.802.630 dan BB= 14.799.900 ≥ 14.797.400, sub-masalah 2, BA=

14.800.740 ≤ 14.802.630 dan BB= 14.797.400 ≥ 14.797.400, serta nilai variabel keputusan sub-masalah 1 dan 2 masih ada yang tidak *integer*, maka sub-masalah 1 dapat dicabangkan menjadi sub-masalah 3 dan 4 dan sub-masalah 2 dapat dicabangkan menjadi sub-masalah 5 dan 6.

Iterasi 2:

Sub-masalah 1 memiliki batas atas (BA) dan batas bawah (BB) sebagai berikut:

BA = 14.802.620 dengan $x_1 = 1281$, $x_2 = 966{,}67$, $x_3 = 1500$, $x_4 = 1752{,}33$

BB = 14.799.900 dengan $x_1 = 1281$, $x_2 = 966$, $x_3 = 1500$, $x_4 = 1752$

Karena pecahan terbesar terdapat pada $x_2 = 966{,}67$ maka tambahan kendala untuk sub-masalah 3 adalah $x_2 \geq 967$ dan untuk sub-masalah 4 adalah $x_2 \leq 966$. Sehingga diperoleh:

1. Sub-masalah 3

   Maksimumkan : (4.1)

   Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \geq 1281$, $x_2 \geq 967$

   Hasil dari metode simpleks adalah sebagai berikut:

   Solusi sub-masalah 3: $x_1 = 1281$, $x_2 = 967$, $x_3 = 1500$, $x_4 = 1752{,}33$ dengan Z = 14.803.541 (**Solusi Tidak Layak)** karena nilai solusi lebih besar dari batas atas.

2. Sub-masalah 4

Maksimumkan : (4.1)

Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \geq 1281, x_2 \leq 966$

Hasil dari metode simpleks adalah sebagai berikut:

Solusi sub-masalah 4: $x_1 = 1281{,}42, x_2 = 966, x_3 = 1500,$

$x_4 = 1752{,}58$ dengan Z = 14.802.430

Iterasi 3:

Sub-masalah 2 memiliki batas atas (BA) dan batas bawah (BB) sebagai berikut:

BA = 14.800.740 dengan $x_1 = 1280$, $x_2 = 966{,}67$, $x_3 = 1500$,
$x_4 = 1752{,}58$

BB = 14.797.400 dengan $x_1 = 1280$, $x_2 = 966$, $x_3 = 1500$,
$x_4 = 1752$

Karena pecahan terbesar terdapat pada $x_2 = 966{,}67$ maka tambahan kendala untuk sub-masalah 5 adalah $x_2 \geq 967$ dan untuk sub-masalah 6 adalah , $x_2 \leq 966$. Sehingga diperoleh:

1. Sub-masalah 5

Maksimumkan : (4.1)

Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \leq 1280, x_2 \geq 967$

Hasil dari metode simpleks adalah sebagai berikut:

Solusi sub-masalah 5: $x_1 = 1280, x_2 = 967, x_3 = 1500,$

$x_4 = 1752{,}58$ dengan Z = 14.801.679 (**Solusi Tidak Layak)** karena nilai solusi lebih besar dari batas atas.

2. Sub-masalah 6

Maksimumkan : (4.1)

Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \leq 1280$, $x_2 \leq 966$

Hasil dari metode simpleks adalah sebagai berikut:

Solusi sub-masalah 6: $x_1 = 1280$, $x_2 = 966$, $x_3 \leq 1500$, $x_4 = 1752{,}58$ dengan Z = 14.798.870

Karena nilai solusi dari sub-masalah 4 dan 6 tidak lebih besar dari nilai batas atas dan tidak lebih kecil dari nilai batas bawah yaitu sub-masalah 4, BA= 14.802.430 ≤ 14.802.620 dan BB= 14.799.900 ≥ 14.797.400, sub-masalah 6, BA= 14.798.870 ≤ 14.800.740 dan BB= 14.797.400 ≥ 14.797.400, serta nilai variabel keputusan sub-masalah 4 dan 6 masih ada yang tidak *integer*, maka sub-masalah 4 dapat dicabangkan menjadi sub-masalah 7 dan 8 sedangkan sub-masalah 6 dapat dicabangkan menjadi sub-masalah 9 dan 10

Iterasi 4:

Sub-masalah 4 memiliki batas atas (BA) dan batas bawah (BB) sebagai berikut:

BA = 14.802.430 dengan $x_1 = 1281{,}42$, $x_2 = 966$, $x_3 = 1500$, $x_4 = 1752{,}58$

BB = 14.799.900 dengan $x_1 = 1281$, $x_2 = 966$, $x_3 = 1500$, $x_4 = 1752$

Karena pecahan terbesar terdapat pada $x_4 = 1752{,}58$ maka tambahan kendala untuk sub-masalah 7 adalah $x_4 \geq 1753$ dan untuk sub-masalah 8 adalah $x_4 \leq 1752$. Sehingga diperoleh:

1. Sub-masalah 7

   Maksimumkan : (4.1)

   Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \geq 1281$, $x_2 \leq 966$, $x_4 \geq 1753$

   Hasil dari metode simpleks adalah sebagai berikut:

   Solusi sub-masalah 7: $x_1 = 1281{,}42$, $x_2 = 966$, $x_3 = 1500$, $x_4 = 1753$ dengan Z = 14.802.450 (**Solusi Tidak Layak)** karena nilai solusi lebih besar dari batas atas.

2. Sub-masalah 8

   Maksimumkan : (4.1)

   Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \geq 1281$, $x_2 \leq 966$, $x_4 \leq 1752$

   Hasil dari metode simpleks adalah sebagai berikut:

   Solusi sub-masalah 8: $x_1 = 1282$, $x_2 = 966$, $x_3 = 1500$, $x_4 = 1752$ dengan **Z = 14.802.620 (Solusi Optimum)**

Iterasi 5:

Sub-masalah 6 memiliki batas atas (BA) dan batas bawah (BB) sebagai berikut:

BA =14.798.870 dengan $x_1 = 1280$, $x_2 = 966$, $x_3 \leq 1500$, $x_4 = 1752{,}58$

BB = 14.797.400 dengan $x_1 = 1280$, $x_2 = 966$, $x_3 = 1500$, $x_4 = 1752$

Karena pecahan terbesar terdapat pada $x_4 = 1752{,}58$ maka tambahan kendala untuk sub-masalah 7 adalah $x_4 \geq 1753$ dan untuk sub-masalah 8 adalah $x_4 \leq 1752$. Sehingga diperoleh:

1. Sub-masalah 9

   Maksimumkan : (4.1)

   Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \leq 1280$, $x_2 \leq 966$, $x_4 \geq 1753$

   Hasil dari metode simpleks adalah sebagai berikut:

   Solusi sub-masalah 9: $x_1 = 1280$, $x_2 = 966$, $x_3 = 1500$, $x_4 \leq 1753$ dengan Z= 14.799.950 (**Solusi Tidak Layak)** karena nilai solusi lebih kecil dari batas bawah.

2. Sub-masalah 10

   Maksimumkan : (4.1)

   Kendala : (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), $x_1 \leq 1280$, $x_2 \leq 966$, $x_4 \leq 1752$

   Hasil dari metode simpleks adalah sebagai berikut:

   Solusi sub-masalah 10: $x_1 = 1280$, $x_2 = 966$, $x_3 = 1500$, $x_4 = 1752$ dengan Z = 14.797.400

Iterasi berhenti karena nilai solusi dari semua sub-masalah sudah berupa bilangan bulat. Selanjutnya dipilih solusi optimal terbaik.

Objective
Maximize
Minimize

Instruction
There are more results available in additional windows. These may be opened by using

Linear Programming Results

(untitled) Solution

	X1	X2	X3	X4		RHS	Dual
Maximize	2500	2800	2950	2550			
Constraint 1	150	150	150	150	<=	900000	0
Constraint 2	30	30	30	30	<=	200000	0
Constraint 3	10	10	10	10	<=	60000	0
Constraint 4	13	13	20	13	<=	100000	0
Constraint 5	10	10	10	10	<=	55000	250
Constraint 6	40	40	40	40	<=	220000	0
Constraint 7	10	10	10	10	<=	60000	0
Constraint 8	56	0	35	0	<=	200000	0
Constraint 9	0	0	60	0	<=	90000	7,5
Constraint 10	0	0	0	97	<=	170000	0
Constraint 11	0	60	0	0	<=	58000	0
NEW Constraint 12	1	0	0	0	>=	1281	0
NEW Constraint 13	0	1	0	0	<=	966	300
NEW Constraint 14	0	0	0	1	<=	1752	50
Solution->	1282	966	1500	1752		14802400	

Gambar 4. 8 Hasil akhir Metode Branch and Bound

Dari hasil perhitungan tersebut, maka diambil sub-masalah dengan nilai optimal terbesar yaitu Z = Rp 14.802.400 dengan setiap jenis roti masing-masing diproduksi yaitu Roti Kasur sebanyak 1282 buah, Roti Kepang diproduksi 966 buah, Roti Bantal diproduksi 1500 buah, Roti Sobek diproduksi 1752 buah.

Dengan keuntungan penjualan sebesar Rp 14.802.400, jumlah roti yang bisa diproduksi adalah 5.500 roti.

F. Perbandingan Keuntungan

Perbandingan keuntungan usaha dagang Safa Bakery dengan keuntungan menggunakan metode *Branch and Bound* dapat dilihat pada Tabel 4.6 sebagai berikut:

Tabel 4.6. Perbandingan Keuntungan Safa Bakery dan Keuntungan dengan Menggunakan Metode *Branch and Bound*

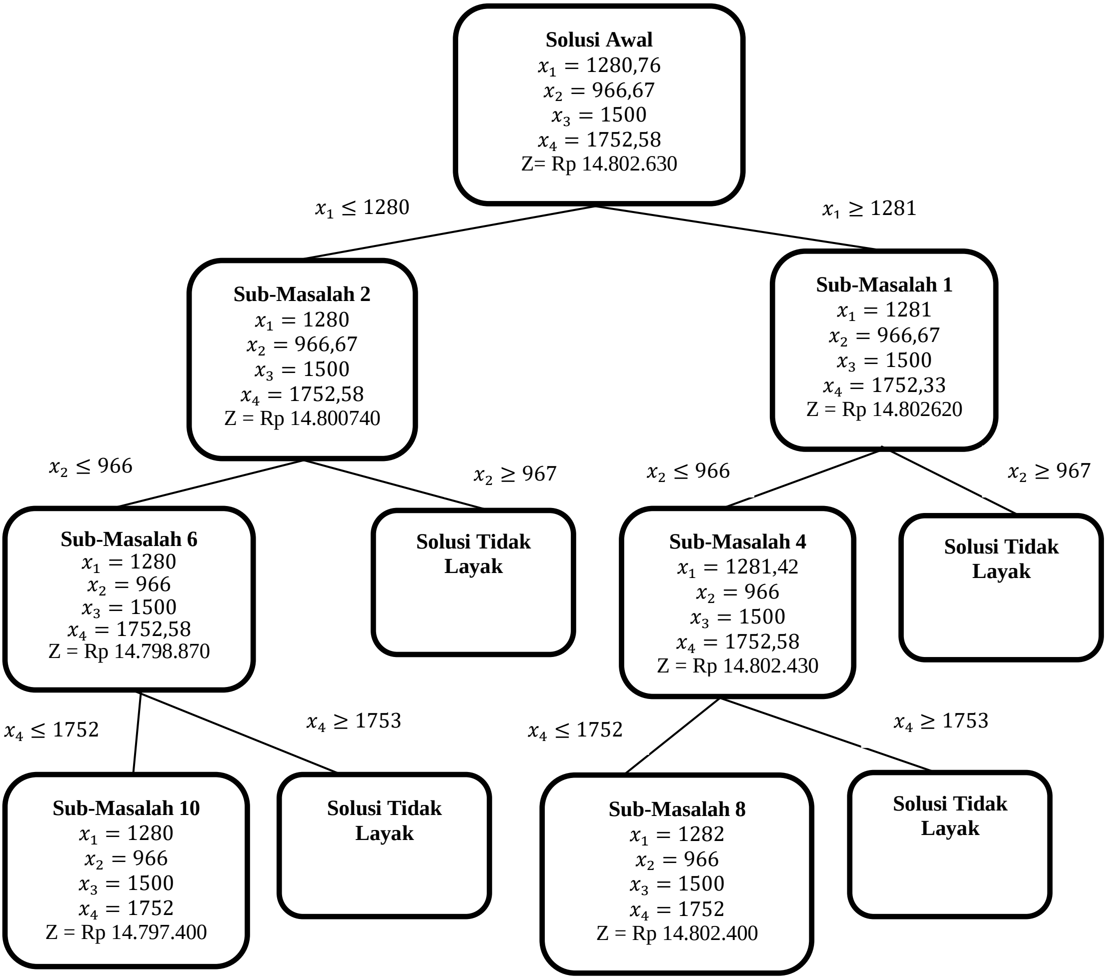
Jenis Roti	Perusahaan		Metode *Branch and Bound*	
	Jumlah Roti diproduksi	Keuntungan	Jumlah Roti diproduksi	Keuntungan
Roti Kasur	1.845	Rp 4.612.500	1.282	Rp 3.205.000
Roti Kepang	920	Rp 2.576.000	966	Rp 2.704.800
Roti Bantal	1.032	Rp 3.044.400	1.500	Rp 4.425.000
Roti Sobek	1.634	Rp 4.166.700	1.752	Rp 4.467.600
Total	5.431	Rp 14.399.600	5.500	Rp 14.802.400

Keuntungan Safa Bakery adalah sebesar Rp 14.399.600 dengan menggunakan metode *Branch and Bound,* keuntungan naik sebedar Rp 14.802.400. Jika keuntungan dibuat dalam persen maka keuntungan yaitu sebesar 2,79% atau Rp 402.800 dari keuntungan Safa Bakery dalam waktu 1 bulan memproduksi 5.500 roti yang terdiri dari Roti Kasur sebanyak 1.282 roti, Roti Kepang sebanyak 966 roti, Roti Bantal sebanyak 1.500 roti dan Roti Sobek sebanyak 1.752 roti.

G. Pohon Penyelesaian Metode *Branch and Bound*

Berikut adalah gambar diagram penyelesaian dengan menggunakan Metode *Branch and Bound:*

Gambar 4.9. Diagram penyelesaian menggunakan Metode *Branch and Bound*



BAB V

SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil dan pembahasan yang dilakukan pada bab 4, maka disimpulkan bahwa:

1. Hasil perhitungan dengan penggunaan metode *Branch and Bound,* banyak roti yang optimal diproduksi dalam waktu satu bulan adalah 5.500 roti yang terdiri dari Roti Kasur sebanyak 1282 roti, Roti Kepang diproduksi sebanyak 966 roti, Roti Bantal diproduksi sebanyak 1500 roti dan Roti Sobek diproduksi sebanyak 1752 roti dengan keuntungan maksimal sebesar Rp 14.802.400,00.
2. Keuntungan usaha dagang Safa Bakery adalah sebesar Rp 14.399.600. Dengan menggunakan metode *Branch and Bound* keuntungan naik sebesar 2,79 % atau Rp 402.800 dari keuntungan Safa Bakery dalam jangka waktu 1 bulan.

B. Saran

Berdasarkan dari hasil penelitian yang dilakukan pada usaha dagang Safa Bakery bisa dijadikan sebagai bahan rujukan dan informasi tambahan bagi Safa Bakery untuk menetapkan kombinasi produksi agar keuntungan yang diperoleh menjadi maksimal. Disarankan untuk mendapatkan variabel keputusan yang bulat dengan solusi maksimal maka pilihlah percabangan dari nilai optimal yang lebih mendekati batas atas.

DAFTAR PUSTAKA

Angeline. dkk. 2014. *Penerapan Metode Branch and Bound dalam Menentukan Jumlah Produksi Optimum pada CV. XYZ.* Saintia Matematika. Vol. 2, No. 2 hal. 139.

Dewi, Sri Desiana Shintya. dkk. 2014. *Analisis Sensitivitas dalam Optimalisasi Keuntungan Produksi Busana dengan Metode Simpleks.* Jurnal Matematika. Vol. 4, No. 2 hal 92-93.

Dimyati, Tjutju dan Ahmad Dimyati. 2006. *Operation Research Model-Model Pengambilan Keputusan.* Bandung: Sinar Baru Algesindo.

Febrianti, Tia dan Erwin Harahap. 2021. *Penggunaan Aplikasi MATLAB Dalam Pembelajaran Program Linear*. Jurnal Matematika Vol. 20, No. 1

Hikmah dan Nursyafitri. 2017. *Aplikasi Integer Programming untuk Meminimumkan Biaya Produksi pada Siaputo Aluminium.* Jurnal Saintifik. Vol. 3, No.2.

Jumarin. 2012. *Optimalisasi Produksi pada Perusahaan Holland Bakery Sudirman Pekanbaru Menggunakan Linier Programming.* Skripsi. Pekanbaru: Universitas Islam Negeri Sultan Syarif Kasim Riau.

Maslihah, Siti. 2015. *Metode Pemecahan Masalah Integer Programming.* Jurnal at-Taqaddum. Volume 7, No.2 hal.216.

Meliana. Dkk. 2019. *Penerapan Algoritma Branch and Boun dalam Menentukan Optimasi Jumlah Produksi Roti (Studi Kasus: CV Sedap Sari Bakery).* Vol. 08, No. 4.

Nurjanah. 2018. *Metode Branch and Bound untuk Meminimalkan Biaya Bahan Baku.* Skripsi. Pekanbaru: Universitan Islam Negeri Sultan Syarif Kasim Riau.

Pagiling, Sahari dan Rais. 2015. *Optimalisasi Hasil Produksi Tahu dan Tempe Menggunakan Metode Branch and Bound.* Jurnal Ilmiah Matematika dan Terapan. Vol. 12, No. 1 hal.53-63.

Purba, Sari Devi dan Faiz Ahyaningsih. 2020. *Integer Programming dengan Metode Branch and Bound dalam Optimasi Jumlah Produks setiap Jenis Roti.* Karismatika. Vol. 6, No. 3. Hal. 3.

Rafflesia, Ulfasari dan Fanani Haryo Widodo. 2014. *Pemrograman Linier.* Bengkulu: Badan Penerbit Fakultas Pertanian UNIB.

Safitri, Elfira. Dkk. 2020. *Penerapan Metode Branch and Bound dalam Optimalisasi Produk Mebel (Studi Kasus: Toko Mebel di Jalan Marsan Panam)*. Jurnal Publikasi Ilmiah Matematika. Vol. 5, No.1. Hal. 46

Siagian, P. 1987. *Penelitian Operasional Universitas Indonesia.* Jakarta: UI-Press.

Siang, Jong Jek. 2011. *Riset Operasi dalam Pendekatan Algoritmis.* Yogyakarta: C.V Andi Yogyakarta.

Sitorus, Parlin. 1997. *Program Linier.* Jakarta: Penerbit Universitas Trisakti.

Supatimah, Sri Siti, dkk. 2019. *Optimasi Keuntungan dengan Metode Branch and Bound.* Jurnal Matematika dan Pendidikan Matematika. Vol. 10, No. 1

Suryawan, Gede. dkk. 2016. Penerapan branch and bound algorithm dalam optimalisasi produksi roti. E-jurnal Matematika. 5 (4): 148-155.

Syaddad, Achmad. 2021. *Analisis Metode Branch and Bound pada Optimalisasi Laba Industri Tambang Batu Bata Putih.* Skripsi. Malang: Universitas Islam Negeri Maulana Malik Ibrahim.

Taha, H. 1997. *Operation Research an Introduction.* New Jersey: Prentice-Hall International.

Ulfiana, Harni. 2009. *Optimsi Jaringan Listrik dengan Algoritma Prim dan Aplikasi Program MATLAB (Studi Kasus PLN Kota Pekalongan)*. Skripsi. Semarang: Universitas Negeri Semarang

DAFTAR RIWAYAT HIDUP

A. Identitas Diri

Nama Lengkap	: Nunu Marwati
Tempat Tanggal Lahir	: Pati, 21 November 1998
Alamat	: Desa Maitan Rt 01 Rw 03 Kec. Tambakromo, Kab. pati
Nomor Hp	: 085326546092
Email	: nunumarwati21@gmail.com

B. Riwayat Pendidikan

Pendidikan Formal

1. SD Negeri Maitan 01 Kabupaten Pati lulus tahun 2010
2. SMP Negeri 02 Tambakromo Kab. Pati lulus tahun 2014
3. MAN Lasem Kabupaten Rembang lulus tahun 2017
4. UIN Walisongo Semarang

Pendidikan Non Formal

1. TPQ Al-Huda Maitan Tambakromo Pati
2. Pondok Pesantren Al-Hamidiyyah Lasem Kabupaten Rembang
3. Pondok Pesantren Putri Tahfidzul Qur'an Al-Hikmah Tugurejo Tugu Semarang

Semarang, 28 Juni 2023

Nunu Marwati
NIM. 1708046019

LAMPIRAN-LAMPIRAN

Lampiran 1.1. Data Produksi Roti pada Safa Bakery bulan Februari 2022

Jenis Roti	Jumlah Produksi
Roti Kasur	1032
Roti Kepang	920
Roti Bantal	1845
Roti Sobek	1634
Total pesanan	5431

Lampiran 1. 2. Data komposisi bahan baku untuk 1 buah roti dan data persediaan bahan baku 4 jenis roti

Bahan Baku Roti	Jenis Roti				Persediaan Maksimal	Satuan
	Roti Kasur	Roti Kepang	Roti Bantal	Roti Sobek		
Terigu	150	150	150	150	900000	Gram
Gula	30	30	30	30	200000	Gram
Telur	10	10	10	10	60000	Gram
Margarin	13	13	20	13	100000	Gram
Garam	10	10	10	10	55000	Gram
Susu Bubuk	40	40	40	40	220000	Gram
Ragi	10	10	10	10	60000	Gram
Coklat	56	0	35	0	200000	Gram
Pisang	0	0	60	0	90000	Gram
Selai	0	0	0	97	170000	Gram

Seres	0	60	0	0	58000	gram

Lampiran 1.3. Data harga jual, biaya produksi dan keuntungan penjualan dari 4 jenis roti

No	Jenis Roti	Harga Penjualan	Biaya Produksi	Keuntungan (Rupiah)
1.	Roti Kasur	Rp 10.000	Rp 7.500	Rp 2.500
2.	Roti Kepang	Rp 10.000	Rp 7.200	Rp 2.800
3.	Roti Bantal	Rp 13.000	Rp 10.050	Rp 2.950
4.	Roti Sobek	Rp 11.000	Rp 8.450	Rp 2.550

Lampiran 1.4. Rincian biaya bahan baku Roti Kasur di Safa Bakery

No	Bahan yang diperlukan	Takaran yang digunakan	Harga	Jumlah
1	Tepung	0,15	8000/kg	1200
2	Gula	0,03	10500/kg	315
3	Telur	0,01	18000/kg	180
4	Margarine	0,013	21000/kg	273
5	Garam	0,01	4000/kg	88
6	Susu Bubuk	0,04	22000/kg	880
7	Ragi	0,01	21000/kg	210
9	Coklat	0,056	26000/kg	1456
10	Biaya Listrik			54
11	Biaya Gas Elpiji			181
12	Kardus			1250
13	Plastik			750
14	Gaji Karyawan			663

	Total Harga			7500

Lampiran 1.5. Rincian biaya bahan baku Roti Kepang di Safa Bakery

No	Bahan yang diperlukan	Takaran yang digunakan	Harga	Jumlah
1	Tepung	0,15	8000/kg	1200
2	Gula	0,05	10500/kg	525
3	Telur	0,01	18000/kg	180
4	Margarine	0,013	21000/kg	273
5	Garam	0,01	4000/kg	88
6	Susu Bubuk	0,04	22000/kg	880
7	Ragi	0,01	21000/kg	210
8	Seres	0,06	20000/kg	1200
9	Biaya Listrik			50
10	Biaya Gas Elpiji			194
11	Kardus			1250
12	Plastik			750
13	Gaji Karyawan			400
	Harga Total			7200

Lampiran 1.6. Rincian biaya bahan baku Roti Bantal di Safa Bakery

No	Bahan yang diperlukan	Takaran yang digunakan	Harga	Jumlah
1	Tepung	0,15	8000/kg	1200
2	Gula	0,03	10500/kg	315
3	Telur	0,01	18000/kg	180
4	Margarine	0,02	21000/kg	420
5	Garam	0,01	8800/kg	88
6	Susu Bubuk	0,04	22000/kg	880

7	Ragi	0,01	21000/kg	210
8	Pisang	0,06	20000/kg	1200
9	Coklat	0,097	26000/kg	2522
10	Biaya Listrik			91
11	Biaya Gas Elpiji			281
12	Kardus			1250
13	Plastik			750
14	Gaji Karyawan			663
	Harga Total			10050

Lampiran 1.7. Rincian biaya bahan baku Roti Sobek di Safa Bakery

No	Bahan yang diperlukan	Takaran yang digunakan	Harga	Jumlah
1	Tepung	0,15	8000/kg	1200
2	Gula	0,03	10500/kg	315
3	Telur	,01	18000/kg	180
4	Margarine	0,013	21000/kg	273
5	Garam	0,01	4000/kg	88
6	Susu Bubuk	0,04	22000/kg	880
7	Ragi	0,01	21000/kg	210
8	Selai	0,097	23000/kg	2231
9	Biaya Listrik			79
10	Biaya Gas elpiji			331
11	Kardus			1250
12	Plastik			750
13	Gaji Karyawan			663
	Total Harga			8450

Lampiran 1.8. Foto-foto pada saat Penelitian