



TESIS - TI185401

**MODEL PERSEDIAAN DAN PENETAPAN HARGA
MULTI-PRODUK PADA *FRESH GOODS* DENGAN
ALGORITMA *ARTIFICIAL BEE COLONY* (ABC)**

**AMALIA RIDHAWATI
02411750030014**

Dosen Pembimbing
NINIET INDAH ARVITRIDA , ST., MSIE., PhD.

**PROGRAM MAGISTER
MANAJEMEN LOGISTIK DAN RANTAI PASOK
DEPARTEMEN TEKNIK SISTEM DAN INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI DAN REKAYASA SISTEM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
2020**



TESIS - TI185401

**MODEL PERSEDIAAN DAN PENETAPAN HARGA
MULTI-PRODUK PADA *FRESH GOODS* DENGAN
ALGORITMA *ARTIFICIAL BEE COLONY* (ABC)**

**AMALIA RIDHAWATI
02411750030014**

**Dosen Pembimbing
NINIET INDAH ARVITRIDA , ST., MSIE., PhD.**

**PROGRAM MAGISTER
MANAJEMEN LOGISTIK DAN RANTAI PASOK
DEPARTEMEN TEKNIK SISTEM DAN INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI DAN REKAYASA SISTEM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
2020**



THESIS - TI185401

**AN INVENTORY AND PRICING MODEL FOR
FRESH GOODS MULTI-PRODUCT WITH
ARTIFICIAL BEE COLONY (ABC) ALGORITHM**

**AMALIA RIDHAWATI
02411750030014**

**Supervisor
NINIET INDAH ARVITRIDA , ST., MSIE., PhD.**

**PROGRAM MAGISTER
MANAJEMEN LOGISTIK DAN RANTAI PASOK
DEPARTEMEN TEKNIK SISTEM DAN INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI DAN REKAYASA SISTEM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
2020**

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh:

AMALIA RIDHAWATI

NRP: 02411750030014

Tanggal Ujian : 15 Agustus 2020

Periode Wisuda : September 2020

Disetujui oleh:

Pembimbing:

Niniet Indah Arvitrida, S.T., M.T., Ph.D.
NIP: 198407052009122007



Penguji:

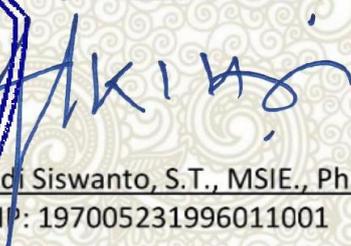
1. Prof. Iwan Vanany, S.T., M.T., Ph.D.
NIP: 197109271999031002
2. Dr. Ir. Bambang Syairudin, M.T
NIP: 196310081990021001



Departemen Teknik Sistem dan Industri
Fakultas Teknologi Industri dan Rekayasa Sistem

Nurhadi Siswanto, S.T., MSIE., Ph.D.

NIP: 197005231996011001



(Halaman ini sengaja dikosongkan)

LEMBAR PERNYATAAN KEASLIAN TESIS

Saya yang bertanda tangan di bawah ini:

Nama : Amalia Ridhawati
NRP : 02411750030014
Program Studi : Magister Teknik Industri - ITS

Menyatakan bahwa tesis dengan judul

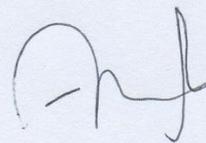
**“MODEL PERSEDIAAN DAN PENETAPAN HARGA MULTI-PRODUK
PADA *FRESH GOODS* DENGAN ALGORITMA *ARTIFICIAL BEE
COLONY (ABC)*”**

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Agustus 2020

Yang membuat pernyataan



Amalia Ridhawati

NRP.02411750030014

(Halaman ini sengaja dikosongkan)

**MODEL PERSEDIAAN DAN PENETAPAN HARGA MULTI-PRODUK
PADA FRESH GOODS DENGAN ALGORITMA ARTIFICIAL BEE
COLONY (ABC)**

Nama : Amalia Ridhawati
NRP : 024 1175 003 0014
Dosen Pembimbing : Niniet Indah Arvitrida, S.T., M.T. Ph.D.

ABSTRAK

Dewasa ini, konsumen lebih sadar akan kesehatan karena standar hidup mereka yang terus meningkat dan membaik. Konsumen lebih suka produk yang masih segar dan memiliki kualitas yang baik. Oleh karena itu, permintaan akan *fresh goods* telah meningkat secara signifikan di tahun-tahun terakhir. Salah satu tempat yang memiliki fungsi untuk memenuhi kebutuhan konsumen tersebut adalah ritel. Pendekatan yang dapat digunakan untuk mendukung optimalisasi peran ritel pada penelitian ini adalah manajemen persediaan dan penetapan harga. Penelitian ini mengembangkan model persediaan dan penetapan harga multi-produk pada *fresh goods* untuk produk independen dan substitusi. Dimana permintaan pada penelitian ini dipengaruhi oleh harga jual, *displayed stock*, dan *expiration date*. Metode yang digunakan dalam penelitian ini adalah metode metaheuristik dengan Algoritma *Artificial Bee Colony* dan bahasa pemrograman *python*. Model tersebut memiliki fungsi tujuan untuk memaksimalkan total *profit* gabungan. Terdapat dua skenario yang dilakukan untuk masing-masing kategori produk (independen dan substitusi). Skenario 1 adalah batasan kapasitas gudang untuk masing-masing produk dan skenario 2 adalah batasan kapasitas gudang untuk seluruh produk/*aggregate*. Hasilnya skenario 2 memiliki nilai total profit gabungan yang lebih besar dibandingkan dengan skenario 1 untuk produk independen maupun substitusi. Hasil analisis sensitivitas didapatkan bahwa semakin tinggi nilai parameter α , β , s , dan w maka semakin tinggi nilai *total profit* gabungan. Sedangkan semakin tinggi nilai parameter c , h , dan o maka semakin kecil nilai *total profit* gabungan. Sedangkan untuk parameter λ , semakin tinggi nilai parameter tersebut maka akan menurunkan *total profit* gabungan produk independen pada skenario 1, produk independen skenario 2, dan produk substitusi skenario 2. Sedangkan untuk produk substitusi skenario 1 mengalami peningkatan *profit*.

Kata kunci: *inventory, pricing, perishable item, artificial bee colony*

(Halaman ini sengaja dikosongkan)

***AN INVENTORY AND PRICING MODEL FOR FRESH GOODS
MULTI-PRODUCT WITH ARTIFICIAL BEE COLONY (ABC) ALGORITHM***

Nama : Amalia Ridhawati
NRP : 024 1175 003 0014
Dosen Pembimbing : Niniet Indah Arvitrida, S.T., M.T. Ph.D.

ABSTRACT

Nowadays, consumers are more health-conscious as their standard of living. Consumers prefer products that are fresh and have good quality. Therefore, the demand for fresh goods has increased significantly in recent years. One of the places that has a function to meet consumer needs is retail. Approaches that can be used to support the optimization of the role of retail in this study are inventory management and pricing. This research develops a multi-product inventory model and pricing of fresh goods for independent and substituted products. The demand in this research is depended on the selling price, displayed stock, and expiration date. The method used in this research is metaheuristic with Artificial Bee Colony Algorithm and Python programming language. The model has an objective function to maximize the total profit. There are two scenarios carried out for each product category (independent and substitution). Scenario 1 is the warehouse capacity limit for each product and scenario 2 is the warehouse capacity limit for all products/aggregates. The result is that scenario 2 has greater total profit value than scenario 1 for both independent and substituted products. The results of the sensitivity analysis show that the higher the value of the parameters α , β , s , and w , the higher the total value of the total profit. Meanwhile, the higher the value of the parameters c , h , and o , the smaller the total profit value. As for the λ parameter, the higher the value of the parameter, the lower the total profit of the independent products in scenario 1, the independent product in scenario 2, and the substitute product in scenario 2. Meanwhile, the profit increases for the substituted product in scenario 1.

Keyword: *inventory, pricing, perishable item, artificial bee colony*

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas limpahan rahmat dan rezeki-Nya sehingga penulis dapat menyelesaikan penulisan Tesis yang berjudul “Model Persediaan dan Penetapan Harga Multi-Produk pada *Fresh Goods* dengan Algoritma *Artificial Bee Colony* (ABC) dengan baik dan tepat waktu. Tak lupa juga shalawat serta salam tercurahkan kepada junjungan Nabi Muhammad SAW yang telah menyampaikan petunjuk kepada umatnya.

Laporan tesis ini diajukan sebagai syarat untuk menyelesaikan studi Strata 2 (S2) di Jurusan Teknik Industri. Selama pelaksanaan dan penyusunan tesis ini, penulis telah menerima bantuan dari berbagai pihak. Oleh sebab itu, pada kesempatan ini, penulis tidak lupa mengucapkan terima kasih dan penghargaan kepada:

1. Allah SWT yang telah memberikan kemudahan, kelancaran dan keyakinan kepada penulis, bahwa penulis mampu menyelesaikan Tesis ini;
2. Suami dan anak tercinta H. Adji Candra Kurniawan, M.T. dan Aslan Muhammad Adji yang selalu mendukung, mendoakan, dan memotivasi penulis;
3. Kedua orang tua penulis yaitu Bapak H. Ahmad Suryani dan Ibu Hj. Saphah, adik penulis yaitu Abdul Hafiz, S.Kom. dan Dinda Bazliah, S.T., serta mertua penulis yaitu Hj. Elok Hari Wahyuningsih dan H. Yusriansyah, yang senantiasa mendoakan dan memotivasi penulis;
4. Ibu Niniet Indah Arvitrida, S.T., M.T. Ph.D. selaku dosen pembimbing penulis yang telah memberikan pengarahan, bimbingan serta dukungan penuh selama penulis menyelesaikan Tesis ini;
5. Bapak Prof. Ir. I Nyoman Pujawan M.Eng., Ph.D. dan Bapak Erwin Widodo, S.T., M. Eng., Dr. Eng selaku penguji Seminar Proposal serta Bapak Prof. Iwan Vanany, S.T., M.T., Ph.D. dan Bapak Dr. Ir. Bambang Syairudin M.T. selaku penguji Seminar Hasil penulis untuk penelitian ini yang telah memberikan saran-saran perbaikan serta bimbingan yang mendukung penelitian ini;

6. Teman – teman Pasca Sarjana Teknik Industri Program Magister dan Doktoral Teknik Industri ITS, yang telah memberikan bantuan dan memotivasi penulis dalam menyelesaikan Tesis ini.

Dalam penulisan Tesis ini, penulis merasa masih banyak kekurangan pada teknis penulisan dan materi laporan. Untuk itu, kritik dan saran dari semua pihak sangat diharapkan demi penyempurnaan pembuatan Tesis ini. Penulis berharap semoga Tesis ini dapat bermanfaat bagi objek amatan dan rekan – rekan di Teknik Industri ITS pada khususnya.

Surabaya, Agustus 2020

Penulis

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vivii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	5
1.3 Tujuan Penelitian	6
1.4 Manfaat Penelitian	6
1.5 Ruang Lingkup Penelitian.....	7
1.5.1 Batasan Penelitian	7
1.5.2 Asumsi Penelitian.....	7
1.6 Sistematika Penulisan.....	8
BAB 2 KAJIAN PUSTAKA.....	11
2.1 Manajemen Persediaan pada Produk <i>Perishable</i>	11
2.1.1 <i>Review</i> Manajemen Persediaan pada Produk <i>Perishable</i>	13
2.1.2 <i>Review Pricing Policies</i> pada Produk <i>Perishable</i>	16
2.1.3 <i>Multi Item</i> pada Manajemen Persediaan dan Harga Produk <i>Perishable</i>	20
2.1.4 <i>Review</i> Manajemen Persediaan dan Penetapan Harga dengan Batasan Kapaitas Gudang	23
2.2 <i>Artificial Bee Colony</i>	25

2.2.1	Perilaku Koloni Lebah Madu	25
2.2.2	<i>Algoritma Artificial Bee Colony</i>	27
2.3	<i>Review</i> Metaheuristik pada Manajemen Persediaan.....	30
2.4	Gap Penelitian.....	32
BAB 3 METODOLOGI PENELITIAN		37
3.1	Pengembangan Model Konseptual	37
3.2	Pengembangan Model Matematika	38
3.3	Verifikasi	40
3.4	Analisis Hasil.....	41
BAB 4 PENGEMBANGAN MODEL		43
4.1	Model Konseptual.....	43
4.2	Model Matematika.....	44
4.2.1.	Model Dasar	44
4.2.2.	Model Penelitian.....	48
4.3	Skenario	51
5.3.1	Skenario 1: Batasan kapasitas gudang untuk masing-masing produk	51
5.3.1	Skenario 2: Batasan kapasitas gudang untuk seluruh produk (aggregate).....	53
4.4	Verifikasi	54
BAB 5 ANALISIS DAN PEMBAHASAN.....		59
5.1.	Parameter dan <i>Constraint, Hyper Parameter, Fitness</i> , dan <i>Algoritma Artificial Bee Colony</i>	59
5.1.1.	Parameter	59
5.1.2.	Tahapan Percobaan Numerik dengan <i>Algoritma Artificial Bee Colony (ABC)</i>	62

5.2.	Percobaan Numerik.....	63
5.3.	Analisis Sensitivitas	68
5.3.1	Analisis Sensitivitas Parameter α	68
5.3.2	Analisis Sensitivitas Parameter β	76
5.3.3	Analisis Sensitivitas Parameter λ	84
5.3.4	Analisis Sensitivitas Parameter c	92
5.3.5	Analisis Sensitivitas Parameter h	100
5.3.6	Analisis Sensitivitas Parameter o	108
5.3.7	Analisis Sensitivitas Parameter s	116
5.3.8	Analisis Sensitivitas Parameter w	124
BAB 6 KESIMPULAN DAN SARAN.....		139
6.1.	Kesimpulan	139
6.2.	Saran.....	140
DAFTAR PUSTAKA		141

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2. 1 Perilaku lebah madu dalam mencari nektar	26
Gambar 3.1 Flowchart metodologi penelitian.....	37
Gambar 3.2 Tahapan pengembangan model.....	40
Gambar 4.1 Ilustrasi Konseptual Model	44
Gambar 4.2 Grafik Sistem Persediaan Feng et al., (2017).....	46
Gambar 4. 3 Parameter Solver	56
Gambar 4. 4 Data Input / Parameter Model	57
Gambar 4. 5 Output yang Dihasilkan.....	57
Gambar 5. 1 Tahapan Percobaan Numerik dengan Algoritma Artificial Bee Colony.....	63
Gambar 5. 2 Grafik Iterasi Fitness Skenario 1 Produk Independen.....	64
Gambar 5. 3 Grafik Iterasi Fitness Skenario 1 Produk Substitusi.....	65
Gambar 5. 4 Grafik Iterasi Fitness Skenario 2 Produk Independen.....	66
Gambar 5. 5 Grafik Iterasi Fitness Skenario 2 Produk Substitusi.....	67
Gambar 5. 6 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Independen Parameter α	69
Gambar 5. 7 Grafik Total Profit Skenario 1 Produk Independen Parameter α	70
Gambar 5. 8 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter α	71
Gambar 5. 9 Grafik Total Profit Skenario 1 Produk Substitusi Parameter α	72
Gambar 5. 10 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter α	73
Gambar 5. 11 Grafik Total Profit Skenario 2 Produk Independen Parameter α ...	74
Gambar 5. 12 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter α	75
Gambar 5. 13 Grafik Total Profit Skenario 2 Produk Substitusi Parameter α	76
Gambar 5. 14 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Independen Parameter β	78
Gambar 5. 15 Grafik Total Profit Skenario 1 Produk Independen Parameter β ...	78

Gambar 5. 16 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter β (dalam 1000).....	80
Gambar 5. 17 Grafik Total Profit Skenario 1 Produk Substitusi Parameter β (dalam 1000).....	80
Gambar 5. 18 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter β	82
Gambar 5. 19 Grafik Total Profit Skenario 2 Produk Independen Parameter β	82
Gambar 5. 20 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter β	84
Gambar 5. 21 Grafik Total Profit Skenario 2 Produk Substitusi Parameter β	84
Gambar 5. 22 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Independen Parameter λ	86
Gambar 5. 23 Grafik Total Profit Skenario 1 Produk Independen Parameter λ	86
Gambar 5. 24 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter λ	88
Gambar 5. 25 Grafik Total Profit Skenario 1 Produk Substitusi Parameter λ	88
Gambar 5. 26 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter λ	90
Gambar 5. 27 Grafik Total Profit Skenario 2 Produk Independen Parameter λ	90
Gambar 5. 28 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter λ	92
Gambar 5. 29 Grafik Total Profit Skenario 2 Produk Substitusi Parameter λ	92
Gambar 5. 30 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Independen Parameter c	94
Gambar 5. 31 Grafik Total Profit Skenario 1 Produk Independen Parameter c	94
Gambar 5. 32 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter c	96
Gambar 5. 33 Grafik Total Profit Skenario 1 Produk Substitusi Parameter c	96
Gambar 5. 34 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter c	98
Gambar 5. 35 Grafik Total Profit Skenario 2 Produk Independen Parameter c	98

Gambar 5. 36 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter c.....	100
Gambar 5. 37 Grafik Total Profit Skenario 2 Produk Substitusi Parameter c	100
Gambar 5. 38 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Independen Parameter h.....	102
Gambar 5. 39 Grafik Total Profit Skenario 1 Produk Independen Parameter h .	102
Gambar 5. 40 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter h.....	104
Gambar 5. 41 Grafik Total Profit Skenario 1 Produk Substitusi Parameter h	104
Gambar 5. 42 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter h.....	106
Gambar 5. 43 Grafik Total Profit Skenario 2 Produk Independen Parameter h .	106
Gambar 5. 44 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter h.....	108
Gambar 5. 45 Grafik Total Profit Skenario 2 Produk Substitusi Parameter h	108
Gambar 5. 46 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Independen Parameter o.....	110
Gambar 5. 47 Grafik Total Profit Skenario 1 Produk Independen Parameter o .	110
Gambar 5. 48 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter o.....	112
Gambar 5. 49 Grafik Total Profit Skenario 1 Produk Substitusi Parameter o	112
Gambar 5. 50 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter o.....	114
Gambar 5. 51 Grafik Total Profit Skenario 2 Produk Independen Parameter o .	114
Gambar 5. 52 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter o.....	116
Gambar 5. 53 Grafik Total Profit Skenario 2 Produk Substitusi Parameter o	116
Gambar 5. 54 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Independen Parameter s.....	118
Gambar 5. 55 Grafik Total Profit Skenario 1 Produk Independen Parameter s..	118
Gambar 5. 56 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter s.....	120

Gambar 5. 57 Grafik Total Profit Skenario 1 Produk Substitusi Parameter s	120
Gambar 5. 58 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter s	122
Gambar 5. 59 Grafik Total Profit Skenario 2 Produk Independen Parameter s ..	122
Gambar 5. 60 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter s	124
Gambar 5. 61 Grafik Total Profit Skenario 2 Produk Substitusi Parameter s	124
Gambar 5. 62 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Independen Parameter w	126
Gambar 5. 63 Grafik Total Profit Skenario 1 Produk Independen Parameter w.	126
Gambar 5. 64 Perbandingan Profit Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter w	128
Gambar 5. 65 Grafik Total Profit Skenario 1 Produk Substitusi Parameter w....	128
Gambar 5. 66 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter w	130
Gambar 5. 67 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Independen Parameter w	130
Gambar 5. 68 Perbandingan Profit Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter w	132
Gambar 5. 69 Grafik Total Profit Skenario 2 Produk Substitusi Parameter w....	132

DAFTAR TABEL

Tabel 2. 1 Review manajemen persediaan pada perishable item.....	13
Tabel 2. 2 Perbandingan penggunaan metode metaheuristik	27
Tabel 2. 3 Perbandingan hasil yang diperoleh dengan menggunakan metode DE, PSO, EA, dan algoritma ABC.....	Error! Bookmark not defined.
Tabel 2. 4 Penelitian terdahulu dan gap penelitian	34
Tabel 4. 1 Data Input Model Untuk Verifikasi.....	55
Tabel 4. 2 Perbandingan Output Model Menggunakan Solver vs Output Jurnal Feng et al., (2017)	56
Tabel 4. 3 Perbandingan Output Model	58
Tabel 5. 1 Data Input Produk Independen Skenario 1	60
Tabel 5. 2 Data Input Produk Substitusi Skenario 1	60
Tabel 5. 3 Data Input Produk Independen Skenario 2	61
Tabel 5. 4 Data Input Produk Substitusi Skenario 2	62
Tabel 5. 5 Hasil Percobaan Numerik Skenario 1 Produk Independen.....	64
Tabel 5. 6 Hasil Percobaan Numerik Skenario 1 Produk Substitusi.....	65
Tabel 5. 7 Hasil Percobaan Numerik Skenario 2 Produk Independen.....	66
Tabel 5. 8 Hasil Percobaan Numerik Skenario 2 Produk Substitusi.....	66
Tabel 5. 9 Perbandingan Profit Per Skenario Per Tipe Produk.....	67
Tabel 5. 10 Output Hasil Analisis Sensitivitas Parameter α	68
Tabel 5. 11 Output Hasil Analisis Sensitivitas Parameter α	70
Tabel 5. 12 Output Hasil Analisis Sensitivitas Parameter α	72
Tabel 5. 13 Output Hasil Analisis Sensitivitas Parameter α	74
Tabel 5. 14 Output Hasil Analisis Sensitivitas Parameter β	77
Tabel 5. 15 Output Hasil Analisis Sensitivitas Parameter β	79
Tabel 5. 16 Output Hasil Analisis Sensitivitas Parameter β	81
Tabel 5. 17 Output Hasil Analisis Sensitivitas Parameter β	83
Tabel 5. 18 Output Hasil Analisis Sensitivitas Parameter λ	85
Tabel 5. 19 Output Hasil Analisis Sensitivitas Parameter λ	87
Tabel 5. 20 Output Hasil Analisis Sensitivitas Parameter λ	89
Tabel 5. 21 Output Hasil Analisis Sensitivitas Parameter λ	91

Tabel 5. 22 Output Hasil Analisis Sensitivitas Parameter c	93
Tabel 5. 23 Output Hasil Analisis Sensitivitas Parameter c	95
Tabel 5. 24 Output Hasil Analisis Sensitivitas Parameter c	97
Tabel 5. 25 Output Hasil Analisis Sensitivitas Parameter c	99
Tabel 5. 26 Output Hasil Analisis Sensitivitas Parameter h.....	101
Tabel 5. 27 Output Hasil Analisis Sensitivitas Parameter h.....	103
Tabel 5. 28 Output Hasil Analisis Sensitivitas Parameter h.....	105
Tabel 5. 29 Output Hasil Analisis Sensitivitas Parameter h.....	107
Tabel 5. 30 Output Hasil Analisis Sensitivitas Parameter o.....	109
Tabel 5. 31 Output Hasil Analisis Sensitivitas Parameter o.....	111
Tabel 5. 32 Output Hasil Analisis Sensitivitas Parameter o.....	113
Tabel 5. 33 Output Hasil Analisis Sensitivitas Parameter o.....	115
Tabel 5. 34 Output Hasil Analisis Sensitivitas Parameter s	117
Tabel 5. 35 Output Hasil Analisis Sensitivitas Parameter s	119
Tabel 5. 36 Output Hasil Analisis Sensitivitas Parameter s	121
Tabel 5. 37 Output Hasil Analisis Sensitivitas Parameter s	123
Tabel 5. 38 Output Hasil Analisis Sensitivitas Parameter w	125
Tabel 5. 39 Output Hasil Analisis Sensitivitas Parameter w	127
Tabel 5. 40 Output Hasil Analisis Sensitivitas Parameter w	129
Tabel 5. 41 Output Hasil Analisis Sensitivitas Parameter w	131

BAB 1

PENDAHULUAN

Bab ini akan menjelaskan pendahuluan yang berisikan latar belakang penelitian, perumusan masalah dalam penelitian, tujuan penelitian, manfaat penelitian, dan sistematika penelitian yang digunakan dalam penelitian ini.

1.1 Latar Belakang

Ritel memiliki fungsi sebagai tempat menyalurkan *supply* dari *supplier*/manufaktur untuk memenuhi permintaan konsumen. Pendekatan yang dapat digunakan untuk mendukung optimalisasi peran ritel adalah manajemen persediaan. Persediaan (*inventory*) merupakan bagian penting dalam rantai pasok karena memiliki kontribusi biaya sekitar 30% dari nilai produk (Daniel & Rajendran, 2005). Persediaan adalah barang yang disimpan untuk memenuhi permintaan (Vrat, 2014). Untuk mencapai manajemen persediaan yang optimal, parameter *supply* (pasokan), *demand* (permintaan), serta biaya perlu diperhitungkan dalam sebuah model (Vrat, 2014). Manajemen persediaan ini sendiri memiliki dua permasalahan utama yang harus dipecahkan. Pertama, kapan perusahaan harus memesan material atau bahan baku untuk persediaan? (*reorder point*). Kedua, berapa banyak bahan baku atau barang yang harus dibeli? (*Economic Order Quantity*). Model persediaan kemudian dikembangkan dengan variabel keputusan *lot size* dan *reorder point*, serta berbagai parameter biaya yaitu *carrying* atau *holding cost*, *shortage* atau *stockout cost*, dan *ordering cost* yang disesuaikan dengan model yang ada (Vrat, 2014). Selain itu, berdasarkan penelitian yang telah dilakukan oleh Rai et al. (2013) pengelolaan persediaan *fresh goods* di ritel akan berdampak langsung pada keuntungan apabila dikelola dengan benar.

Dewasa ini, konsumen lebih sadar akan kesehatan karena standar hidup mereka yang terus meningkat dan membaik. Konsumen lebih suka produk yang masih segar dan memiliki kualitas yang baik. Oleh karena itu, permintaan akan *fresh goods* telah meningkat secara signifikan di tahun-tahun terakhir (L. Feng, Chan, & Cárdenas-Barrón, 2017). *Fresh goods* (produk segar) adalah salah satu

jenis produk *perishable* yang akan mengalami kerusakan atau penurunan kualitas sampai periode waktu tertentu. Beberapa contoh produk yang tergolong *fresh goods*, yaitu: sayur-sayuran, buah-buahan, roti, susu, daging, makanan laut, dan lain-lain (L. Feng et al., 2017). Penelitian terdahulu seperti yang telah dilakukan oleh (L. Feng et al., 2017) mengembangkan model persediaan dan penetapan harga pada produk *perishable* khususnya *fresh goods* dengan mempertimbangkan harga dalam mengelola persediaan. Dalam penelitian tersebut, permintaan (*demand*) konsumen tergantung pada harga jual, *displayed volume*, dan *expiration date*. Penelitian tersebut menganalisis solusi optimal pada ritel dengan variabel keputusan, yaitu: *ending stock* (persediaan akhir), *selling price* (harga jual), dan *cycle time* (waktu siklus). Hal ini bertujuan untuk meningkatkan keuntungan (*profit*) dan daya saing dalam rantai pasok produk *perishable* (L. Feng et al., 2017). Penelitian dengan mempertimbangkan harga juga dilakukan oleh (Banerjee & Agrawal, 2017). Penelitian ini bertujuan untuk memaksimalkan keuntungan produk *perishable* dengan mempertimbangkan permintaan yang tergantung pada *freshness* dan harga, serta mempertimbangkan diskon dan kebijakan pemesanan dalam pengoptimalannya. Berbeda dengan penelitian yang dilakukan oleh (L. Feng et al., 2017), penelitian ini mengizinkan adanya *shortage* dengan asumsi *lost sales*. Pada penelitian ini, kesegaran produk (*freshness*) memiliki pengaruh tinggi terhadap aspek ekonomi. Sedangkan penurunan *freshness* menyebabkan penurunan penjualan apabila produk masih dijual dengan harga yang sama. Selain *freshness*, permintaan untuk *fresh goods* juga tergantung pada *expiration date* dan ukuran *shelf space* karena dapat menarik lebih banyak konsumen untuk membeli produk. Wu et al. (2016) melakukan penelitian mengenai manajemen persediaan untuk *fresh goods* dengan asumsi permintaan pada *fresh goods* adalah fungsi *time-varying* dari *freshness*, *displayed volume*, dan *expiration date*. Dengan permintaan yang tergantung pada *freshness* dan *stock*, akan menguntungkan apabila mempertahankan level persediaan yang tinggi di akhir *replenishment cycle*. Oleh karena itu, *zero ending inventory* diubah menjadi *non-zero ending inventory*. Selain itu, Taylor et al. (2014) mengembangkan model *replenishment policy* dan *dynamic pricing* pada produk *perishable* dengan tingkat persediaan tergantung pada permintaan yang dipengaruhi oleh *stock quantity display* serta harga jual

produk. Tujuan dari penelitian ini adalah untuk memaksimalkan keuntungan yang diperoleh dengan menentukan kombinasi *variable* menggunakan prinsip “*pontryagin*” *maximum*. Abad (1996) juga memformulasikan model *dynamic pricing* dan *lot-sizing* pada *reseller* yang menjual produk *perishable*. Abad mengasumsikan aktivitas *backorder* dapat dilakukan selama hal tersebut menguntungkan. Disamping itu, (Mishra, 2017) mengembangkan model mengenai kebijakan optimal untuk produk *perishable* ketika permintaan tergantung pada kesegaran *displayed stock* dan harga jual. Pada penelitian tersebut, *displayed stock* memiliki peran yang efektif dalam meningkatkan permintaan karena mendorong pelanggan untuk membeli lebih banyak produk. Penelitian lain yang membahas *pricing* sekaligus persediaan dilakukan oleh (Valliathal & Uthayakumar, 2011) dengan mengembangkan model EOQ untuk *non-instantaneous deteriorating item* dan mengizinkan adanya *shortages*. Penelitian ini dilakukan pada produk *perishable* seperti makanan, obat-obatan, kimia, dan darah dengan tujuan untuk mengoptimalkan penentuan harga dan kebijakan *replenishment*.

Penelitian terdahulu yang telah dibahas di atas merupakan penelitian untuk *single item*. Berbeda dengan penelitian sebelumnya, Taleizadeh et al. (2010) melakukan penelitian pada *multi items* produk *perishable* dengan tujuan untuk memaksimalkan keuntungan dalam *planning horizon* yang telah ditentukan. Penelitian ini juga mengizinkan adanya *shortage* dengan asumsi *backorder* dan *lost sales*. Pada penelitian ini, dua solusi algoritma metaheuristik yaitu *Genetic Algorithm* (GA) dan *Simulated Annealing* (SA) dilakukan untuk memecahkan masalah tersebut. Hasilnya, metode *Genetic Algorithm* (GA) menghasilkan nilai fungsi objektif yang lebih baik dibandingkan dengan *Simulated Annealing* (SA). Selain itu, penelitian *multi items* produk *perishable* juga dilakukan oleh Shavandi et al. (2012) yang mengembangkan model *single item* (Abad, 2003). Penelitian ini mengembangkan model *pricing* dan *lot sizing* dengan mengkategorikan produk ke dalam 3 (tiga) kategori, yaitu: *independent* (independen), *substitute* (substitusi), dan *complimentary* (pelengkap). Harga dari produk independen tidak memberikan efek pada permintaan produk lain, tetapi harga dari produk sibtitusi dan pelengkap mempengaruhi penetapan harga dan permintaan produk lainnya. Oleh karena itu,

Shavandi et al. (2012) mendefinisikan 3 (tiga) fungsi permintaan sesuai dengan kategori produk, serta mempertimbangkan *budget* (modal), *warehouse* (gudang), dan harga sebagai *constraint* (pembatas). Kemudian, metode *Genetic Algorithm* (GA) digunakan untuk memecahkan model tersebut.

Disinggung sebelumnya oleh Taleizadeh et al. (2010) dan Shavandi et al. (2012), metaheuristik digunakan sebagai metode penyelesaian masalah persediaan. Selain itu, Saracoglu et al. (2014) melakukan penelitian tentang pendekatan Algoritma Genetika untuk *multi product multi period* pada model persediaan *continuous review*. Hasil yang diperoleh menggunakan metode Algoritma Genetika memiliki kualitas yang baik dengan waktu komputasi yang rasional dibandingkan dengan menggunakan *Integer Linear Programming* pada Lingo. (Bhunia & Pareek, 2017) juga menggunakan pendekatan metaheuristik dalam mencari nilai optimal keuntungan pada model persediaan untuk *single deteriorating item*. Pada penelitian tersebut, variabel permintaan tergantung pada strategi pemasaran dan *displayed stock*. Metode metaheuristik yang digunakan adalah Algoritma Genetika dan *Particle Swarm Optimization* dengan nilai hasil Algoritma Genetika yang lebih baik dibandingkan *Particle Swarm Optimization*. Selain itu, (Karaboga & Basturk, 2008) telah melakukan penelitian terhadap kinerja *Artificial Bee Colony (ABC) algorithm* yang merupakan salah satu jenis metaheuristik dan dibandingkan dengan metaheuristik lain yaitu: *Differential Evolution* (DE), *Particle Swarm Optimization* (PSO) dan *Evolutionary Algorithm* (EA) atau *Genetic Algorithm* (GA). Hasil dari perbandingan tersebut adalah *ABC algorithm* dapat memecahkan masalah secara efisien.

Berdasarkan penelitian sebelumnya, belum ada penelitian yang dilakukan mengenai model persediaan dan penetapan harga *multi items* pada produk *perishable* di ritel dengan permintaan tergantung pada harga jual, *displayed stock*, dan *expiration date*. Sehingga, penulis menemukan celah (*gap*) penelitian yaitu mengembangkan model persediaan *single item* pada produk *perishable* yang diadopsi dari (L. Feng et al., 2017) menjadi *multi items* dengan fungsi permintaan yang diadopsi dari model yang dikembangkan oleh Shavandi et al. (2012). Fungsi permintaan dari model (L. Feng et al., 2017) dikembangkan berdasarkan dua kategori produk dari model Shavandi et al. (2012), yaitu: *independent*

(independen) dan *substitute* (substitusi) dengan hasil akhir yang diharapkan adalah mengetahui *total profit* (keuntungan) produk. Penelitian ini tidak dilakukan pada kategori produk komplementer karena *fresh goods* tidak masuk ke dalam kategori produk tersebut. Setelah itu, 2 skenario dilakukan terhadap masing-masing kategori tersebut, dengan skenario 1 batasan kapasitas gudang untuk masing-masing produk dan skenario 2 adalah batasan kapasitas gudang untuk seluruh produk/*aggregate*. Kemudian, *total profit* dari setiap skenario dibandingkan untuk mengetahui skenario mana yang lebih menguntungkan. Setelah itu, analisis sensitivitas dilakukan terhadap parameter model untuk mengetahui efek perubahan parameter terhadap nilai *profit*. Untuk mengetahui metode yang akan digunakan pada penelitian ini, peneliti telah mencari nilai maksimal pada pengembangan model tersebut menggunakan *excel solver*. Namun, solusi optimal yang *feasible* masih belum dapat ditemukan. Selain itu, penulis juga telah menggunakan *software "lingo"* untuk mencari solusi optimal. Namun, waktu yang dibutuhkan sangatlah lama. Penelitian ini juga masuk ke dalam model *non-linear programming* dengan variabel kombinatorial sehingga metode metaheuristik dipilih dalam penyelesaian masalah. Berdasarkan penelitian sebelumnya yang telah disebutkan di atas, *Artificial Bee Colony (ABC) algorithm* merupakan metode metaheuristik yang dipilih untuk mencari nilai maksimal keuntungan karena dapat memecahkan masalah secara efisien. Oleh karena itu, penulis mengambil judul "Model persediaan dan penetapan harga multi-produk pada *fresh goods* dengan *Algoritma Artificial Bee Colony (ABC)*".

1.2 Perumusan Masalah

Permasalahan yang diangkat dalam penelitian ini adalah mengembangkan model persediaan dan penetapan harga *multi-product* pada produk *perishable*. Pada penelitian ini, permintaan tergantung pada harga, *displayed stock*, dan *expiration date* berdasarkan 2 (dua) kategori produk, yaitu: *independent* (independen) dan *substitute* (substitusi). Terdapat 2 skenario yang akan dilakukan terhadap 2 kategori tersebut, yaitu: batasan kapasitas gudang untuk masing-masing produk (skenario 1) dan batasan kapasitas gudang untuk seluruh produk /*aggregate* (skenario 2). Metode penyelesaian penelitian ini menggunakan

pendekatan metaheuristik dengan algoritma yang digunakan adalah *Artificial Bee Colony* (ABC). Selain itu, variabel keputusan yang dicari pada penelitian ini adalah harga jual, *order quantity*, dan *cycle time* dengan fungsi objektif memaksimalkan *total profit* produk *perishable*. Pada penelitian ini juga akan dilakukan analisis sensitivitas terhadap parameter untuk mengetahui efek perubahan parameter terhadap nilai *profit*.

1.3 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai adalah:

1. Mengembangkan model persediaan dan penetapan harga *multi items* pada produk *perishable* yang dapat membantu dalam membuat keputusan yang terkait dengan:
 - a. Menentukan harga optimal produk *perishable*.
 - b. Menentukan optimal *order quantity* produk *perishable* dalam unit.
 - c. Menentukan optimal *cycle time* produk *perishable* dalam unit.
 - d. Menentukan optimal maksimum *profit* produk *perishable*.
2. Membandingkan skenario untuk mengetahui nilai *total profit* yang lebih baik.
3. Melakukan analisis sensitivitas parameter-parameter model terhadap nilai *variable response*.

1.4 Manfaat Penelitian

Manfaat yang akan diperoleh dari dilakukannya penelitian ini adalah:

1. Model yang dikembangkan dapat dijadikan acuan dalam penentuan keputusan terkait produk *perishable* di ritel.
2. Model yang dikembangkan dapat dijadikan bahan pertimbangan dalam menentukan keputusan terbaik terkait maksimalisasi *profit* produk *perishable* di ritel dan penanganan isu terkait pengelolaan persediaan.
3. Model yang dikembangkan diharapkan dapat menjawab permasalahan terkait persediaan produk *perishable* di ritel.
4. Model yang dikembangkan dapat dijadikan referensi untuk penelitian terkait berikutnya.

1.5 Ruang Lingkup Penelitian

Pada sub bab ruang lingkup penelitian, akan dibahas mengenai batasan dan asumsi yang digunakan pada penelitian ini. Tujuan dari ditentukannya ruang lingkup ini adalah untuk menjaga agar penelitian ini memiliki *boundary* yang jelas.

1.5.1 Batasan Penelitian

Batasan yang digunakan dalam penelitian ini adalah:

1. Penelitian dilakukan untuk produk *perishable* khususnya *fresh goods*.
2. Penelitian dilakukan pada model persediaan produk *perishable*.
3. Harga per unit dipertimbangkan dalam model persediaan.
4. Biaya persediaan yang dipertimbangkan adalah *purchasing cost*, *ordering cost*, dan *holding cost*.
5. Penelitian ini dilakukan untuk *multi items*.

1.5.2 Asumsi Penelitian

Asumsi yang digunakan dalam penelitian ini adalah:

1. *Demand rate* adalah *deterministic* dan eksponensial terhadap harga dan merupakan bentuk *multiplicative* (perkalian) dari harga, *freshness*, dan *displayed stock*.
2. *Cycle time* harus kurang atau sama dengan *expiration date* produk.
3. *Ending inventory* harus lebih besar atau sama dengan 0 (*non-negative inventory*).
4. Pada periode *replenishment cycle* persediaan mencapai tingkat *ending inventory E*, ritel menjual seluruh unit *E* dengan *salvage price s* untuk setiap unit.
5. *Order quantity* harus lebih besar atau sama dengan ukuran rak dalam unit dan kurang atau sama dengan kapasitas gudang.
6. Kategori produk dibagi menjadi 2 (dua), yaitu: *independent* (independen) dan *substitute* (substitusi).

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini dibagi menjadi beberapa bab. Adapun penjelasan dari masing-masing bab adalah sebagai berikut:

- **BAB I PENDAHULUAN**

Bab ini membahas tentang latar belakang dilakukannya penelitian, perumusan masalah yang dibahas pada penelitian, ruang lingkup yang digunakan dalam penelitian, tujuan dan manfaat yang bisa diambil dari penelitian yang dilakukan, serta sistematika yang diterapkan dalam penelitian ini.

- **BAB II TINJAUAN PUSTAKA**

Bab ini berisi penjelasan berbagai teori yang digunakan pada pelaksanaan penelitian ini. Teori yang digunakan bersumber dari berbagai referensi seperti jurnal dan buku. Penelitian-penelitian yang sudah dilakukan di area yang sama dengan penelitian ini menjadi landasan teori dalam penulisan penelitian ini.

- **BAB III METODE PENELITIAN**

Bab ini membahas tentang tahapan yang dilakukan dalam penelitian untuk mengembangkan model persediaan dan harga produk *perishable*. Metodologi penelitian ini berguna sebagai panduan dalam melakukan penelitian, sehingga penelitian berjalan secara efektif dan sistematis.

- **IV BAB PENGEMBANGAN MODEL**

Bab ini membahas mengenai model yang digunakan pada penelitian ini untuk mendapatkan hasil yang sesuai dengan rumusan masalah serta rencana pengembangan model dan formulasi matematis.

- **BAB V PERCOBAAN NUMERIK**

Bab ini menjelaskan tentang langkah verifikasi model menggunakan percobaan numerik. Percobaan numerik dilakukan untuk menentukan *output* dari model, serta analisis sensitivitas.

- **BAB VI KESIMPULAN DAN SARAN**

Bab ini membahas tentang penarikan kesimpulan dari penelitian yang dilakukan serta memberikan saran berdasarkan penelitian yang dilakukan dan rekomendasi untuk penelitian selanjutnya.

(Halaman ini sengaja dikosongkan)

BAB 2

KAJIAN PUSTAKA

2.1 Manajemen Persediaan pada Produk *Perishable*

Kebijakan persediaan pada dasarnya dirancang untuk mengantisipasi permintaan yang tidak terduga dimasa datang. Sehingga, wajar apabila beberapa model persediaan mengasumsikan bahwa produk yang disimpan tidak akan mengalami perubahan dari sisi kualitas yang akan mengurangi minat konsumen untuk membeli produk tersebut. Namun, pada kenyataannya terdapat tipe produk yang mengalami perubahan selama penyimpanan dan berdampak langsung pada minat beli konsumen. Produk ini dikenal dengan sebutan produk *perishable*.

(Nahmias, 1982) Mengklasifikasikan produk *perishable* ke dalam dua kategori, yaitu: *fixed lifetime* dan *random lifetime* yang akan dibahas lebih lanjut pada sub-bab berikutnya. Kasus-kasus mengenai produk *perishable* menjelaskan jika *lifetime* diketahui sebagai *random variable* yang memiliki probabilitas. *Variable lifetime* merupakan *variable* yang memiliki pengaruh kuat pada minat belanja konsumen. Hal ini dikarenakan konsumen hanya ingin membeli produk yang memiliki kualitas baik (produk berada di masa awal *lifetime*-nya). Pandangan konsumen terhadap kualitas suatu produk tentunya berbeda antara konsumen satu dengan yang lainnya. Hal ini dipengaruhi oleh kebutuhan, kegiatan, dan kebiasaan masing-masing konsumen (Kardes et al., 2011). Dampak dari perbedaan ini adalah permintaan yang semakin dinamis. Oleh karena itu, untuk merespon dinamika permintaan dari konsumen maka perlu adanya perencanaan dan pengembangan model kebijakan penyimpanan agar dapat mengantisipasi permintaan dan hal yang dapat merugikan (*shortage*, penurunan *service level*) dimasa depan.

Berdasarkan klasifikasi produk *perishable* oleh (Nahmias, 1982), *fresh goods* masuk ke dalam kategori *random lifetime (exponential decay)*. Pengklasifikasian produk *perishable* tersebut secara lebih detil adalah:

a. *Fixed Life Perishability*

Asumsi untuk kategori *fixed life perishability* adalah unit produk dipertahankan dalam bentuk persediaan untuk memuaskan permintaan pelanggan

selama spesifik *fixed time* tertentu. Setelah *fixed time* tersebut, produk harus dibuang. Model ini mengasumsikan bahwa semua unit produk yang tidak kadaluwarsa memiliki utilitas yang sama.

1. Permintaan Deterministik

Model *perishable inventory* deterministik memiliki karakteristik yang umumnya pemesanan dilakukan dalam *optimal policy* memungkinkan produk tidak sampai batas kadaluwarsa. Model *Economic Order Quantity* (EOQ) paling sederhana untuk tipe permintaan ini adalah:

$$Q^* = \sqrt{\frac{2DS}{H}} \quad (2.1)$$

D adalah permintaan; S adalah biaya pesan; dan H adalah biaya penyimpanan.

2. Permintaan Stokastik

Model *perishable inventory* stokastik merupakan tipe yang cukup kompleks. Salah satu kasus yang biasa ditemui, yaitu: ketika level persediaan di-*review* secara periodik dan persediaan dapat disimpan tidak lebih dari 1 (satu) periode. Jika usia produk benar-benar satu periode, keputusan pemesanan untuk periode berikutnya secara berturut-turut adalah independen (Arroow et al., 1958).

b. *Random Lifetime Models*

Pada beberapa model persediaan, *lifetime* stok unit produk tidak dapat ditentukan secara pasti. Unit produk akan dibuang ketika kadaluwarsa dan waktu kadaluwarsa juga tidak dapat ditentukan. Contoh produk untuk kategori *random lifetime* adalah *fresh goods*. Dalam *random lifetime* terdapat satu proses yang muncul dari masalah persediaan produk *perishable* dengan *random lifetime*. Misal *lifetime* dari suatu unit adalah variabel acak dengan distribusi eksponensial yang bernilai negatif dan parameter θ . Jika $I(t)$ adalah jumlah unit yang bertahan sampai waktu t khusus untuk memenuhi permintaan dan jika probabilitas unit dari waktu tambahan s adalah $e^{-\theta s}$ maka jumlah unit yang bertahan sampai waktu $t+s$ adalah variabel acak binomial dengan parameter: $n=I(t)$ dan $p=e^{-\theta s}$ serta jumlah unit yang diharapkan sampai waktu $t+s$ adalah np atau $I(t)\exp\{-\theta s\}$. Oleh karena itu, hal ini dikenal dengan proses *exponential decay*.

Exponential decay juga dapat diturunkan dengan mengasumsikan bahwa *fixed fraction* dari *stock on hand* berkurang pada setiap periode mengikuti distribusi usia persediaan. Misal $I(t)$ merupakan level persediaan pada waktu t dan permintaan diabaikan, asumsi *decay* adalah $I(t+1)=\gamma I(t)$ dengan $0<\gamma<1$ adalah konstan. Kemudian digeneralisasi menjadi $I(t+s)=\gamma^s I(t)$ dengan $\theta = -\ln \gamma$.

2.1.1 Review Manajemen Persediaan pada Produk *Perishable*

Pada sub bab ini akan dijelaskan penelitian yang telah dilakukan dalam manajemen persediaan pada produk *perishable*. Penjelasan mengenai manajemen persediaan mengarah pada produk *perishable* khususnya *fresh goods* di ritel. Rangkuman penelitian dapat dilihat pada tabel 2.1.

Tabel 2. 1 Review manajemen persediaan pada produk *perishable*

Reference	Tujuan Penelitian	Objek Penelitian	Kesimpulan
Padmanabhan dan Vrat (1993)	Memaksimalkan <i>total profit</i> pada produk <i>perishable</i> dengan <i>backlogging</i>	Jenis produk: produk <i>perishable</i> lokasi: manufaktur	Model pada penelitian ini dapat digunakan untuk menentukan <i>optimal inventory policy</i> seperti <i>super market, stationery stores, dan fancy items</i>
Chang dan Dye (1999)	Mengetahui efek <i>backlogging rate</i> pada EOQ untuk meminimalkan biaya <i>inventory</i>	Jenis produk: produk <i>perishable</i>	<i>Backlogging rate</i> dipertimbangkan sebagai fungsi menurun saat waktu tunggu pada <i>replenishment</i> berikutnya
Wu et al (2005)	Menentukan optimal <i>replenishment policy</i> pada <i>non-instantaneous deteriorating items</i> dengan <i>stock-dependent demand</i>	Jenis produk: produk <i>perishable</i> lokasi: manufaktur	Model yang dikembangkan dapat digunakan untuk mengontrol persediaan pada <i>non-instantaneous deteriorating items</i> tertentu
Rai et al (2013)	Meminimalkan kerugian dan memaksimalkan <i>organizational value</i> pada <i>inventory policy</i>	Jenis produk: produk <i>perishable</i> lokasi: ritel	Tingkat permintaan memiliki hubungan yang kuat terhadap <i>perishability</i> , sedangkan <i>freshness</i> dan kondisi fisik produk memiliki hubungan yang lemah
Hidayat dan Fauzi (2015)	Menentukan keputusan optimal dalam <i>procurement inventory system</i> farmasi untuk item yang memiliki <i>expired date</i>	Jenis produk: produk <i>perishable</i> lokasi: manufaktur	Algoritma yang dikembangkan mampu mengurangi 44,61% biaya persediaan jika dibandingkan dengan pendekatan tradisional
Dobson et al (2017)	Mengoptimalkan <i>length</i> dan <i>profit cycle</i>	Jenis produk: produk <i>perishable</i> lokasi: ritel	Model <i>perishable</i> memiliki biaya penyimpanan yang lebih besar daripada produk lainnya

Kirci et al (2018)	Meminimalkan <i>unit cost</i> pada <i>finished goods</i>	Jenis produk: produk <i>perishable</i> lokasi: manufaktur dan ritel	Usia bahan baku yang pendek secara signifikan meningkatkan biaya manufaktur dan ritel.
Khurana (2018)	Mengembangkan model persediaan untuk <i>deteriorating product</i>	Jenis produk: produk <i>perishable</i> lokasi: ritel	Peningkatan dan penurunan <i>demand rate</i> akan mengikuti <i>on-hand inventory level</i>

Salah satu masalah yang paling penting dan sering dihadapi di sebuah perusahaan adalah mengontrol dan menjaga persediaan. Salah satu cara untuk mengatasi masalah ini adalah dengan mengembangkan matematika model dari penelitian-penelitian terdahulu yang memiliki karakteristik masalah yang hampir sama. Dobson et al. (2017) meneliti manajemen persediaan untuk produk *perishable* yang dijual di ritel. Dobson memformulasikan dan menganalisis model *Economic Order Quantity (EOQ)* dengan mempertimbangkan bahwa konsumen lebih mementingkan kualitas daripada usia produk, dan mengasumsikan bahwa *demand rate* berbanding lurus dengan penurunan fungsi usia produk. Penelitian ini termotivasi dari peningkatan minat pelanggan pada *fresh goods* di ritel. Faktanya adalah hal ini tidak hanya terjadi untuk *fresh goods*, tetapi juga untuk *packaged good*. Meskipun pelanggan tidak secara langsung memperhatikan *freshness* pada *packaged good*, pelanggan yang mementingkan kesehatan akan memperhatikan *expiration date* pada produk. Oleh karena itu, Dobson memfokuskan tujuan penelitian pada optimalisasi *cycle length* dan *profit* serta membandingkan model yang dibuatnya dengan model tradisional EOQ. Masalah persediaan lain untuk produk *perishable* yang terjadi pada sudut pandang ritel diteliti oleh (Rai et al., 2013). Rai dan kawan-kawan mengembangkan model EOQ untuk menganalisis data yang telah terkumpul dan menentukan *ordering policy*. Penelitian ini mencoba memodelkan persediaan pada ritel dengan tujuan meminimalkan kerugian dan memaksimalkan keuntungan. Pengambilan keputusan pada penelitian ini mempertimbangkan *self space* dan *reordering item* berdasarkan penggunaan persediaan. Aspek lain yang dipertimbangkan dalam penelitian ini adalah ketersediaan produk pada musim tertentu serta efek dari musim itu sendiri. Ketersediaan produk *perishable* di ritel memiliki sifat permintaan yaitu *non-deterministic* sehingga persediaan harus dikelola

berdasarkan *order quantity*. Kırıcı et al. (2018) melakukan penelitian pada dua eselon rantai pasok dengan *upstream* manufaktur dan *downstream* ritel pada *single* produk *perishable*. Manufaktur mengolah bahan baku menjadi barang jadi yang akan dibeli oleh ritel di setiap *replenishment cycle*. Bahan baku di manufaktur memiliki tingkat *perishability* yang tinggi (yaitu akan kadaluwarsa dalam hitungan jam atau hari), sedangkan barang jadi di ritel mengalami ketidakpastian permintaan dan keusangan. Penelitian ini memodelkan hubungan manufaktur dan ritel sebagai *stackelber game* dimana ritel adalah *leader* yang menentukan *replenishment cycle* untuk meminimalkan biaya per unit pada barang jadi. Hasilnya menunjukkan bahwa usia bahan baku dan barang jadi yang dihubungkan melalui durasi *replenishment cycle* memiliki dampak yang signifikan terhadap biaya. Selain itu, tingginya *mark up* yang dilakukan manufaktur meningkatkan biaya pada ritel.

Penelitian lain mengenai model persediaan untuk produk *perishable* dengan mengizinkan adanya *shortages* diteliti oleh (Khurana, 2018). Penelitian tersebut mengasumsikan permintaan sebagai fungsi waktu dan *available stock level* karena menganggap *available stock level* dapat menarik pelanggan.

Pada penelitian ini, Khurana mengizinkan *shortages* dan *partially backlogging* dengan mengembangkan *backlogging rate* pada dua kasus yang berbeda. Pada kasus pertama, *backlogging rate* diasumsikan konstan, sedangkan kasus lainnya mengasumsikan *backlogging rate* sebagai *time dependent*. Dalam model tradisional EOQ, *shortages* sering diasumsikan benar-benar *backlogged* atau benar-benar *lost*. Namun, beberapa sistem persediaan mengasumsikan bahwa *backlogging rate* bergantung pada lamanya waktu tunggu untuk *replenishment* selanjutnya. Semakin lama waktu tunggu maka semakin kecil *backlogging rate*. (Chang & Dye, 1999) melakukan penelitian terhadap persediaan produk *perishable* yang fokus pada efek *backlogging rate* pada penentuan EOQ. Sehingga, penelitian tersebut mengasumsikan bahwa *demand rate* merupakan fungsi *time-continuous* dan produk mengalami deteriorasi pada *constant rate* dengan *partial backlogging*. Hidayat & Fauzi (2015) mengembangkan algoritma pada model persediaan untuk menentukan pengadaan (*procurement*) yang optimal dalam sistem persediaan farmasi Usaha Mikro Kecil Menengah (UMKM) untuk

item yang memiliki *expired date*. Hal ini dilakukan untuk memperbaiki indikasi kebijakan persediaan yang tidak layak karena adanya produk yang telah melewati *expiration date*. Hal ini juga ditunjukkan oleh penurunan keuntungan pada *Return of Investment* (ROI) yang dipengaruhi oleh *inventory turnover*. Penentuan pengadaan dipengaruhi oleh *lot size*, durasi antar pesanan (*replenishment time*), dan *reorder point*. Hasilnya, Hidayat dan Fauzi membuktikan bahwa algoritma yang dikembangkan mereka mampu mengurangi *total* biaya persediaan sampai 44,61% dari pendekatan tradisional.

Salah satu aspek lain yang mempengaruhi model persediaan adalah *selling rate*. (Vrat, 1993) mengembangkan model persediaan untuk produk *perishable* dengan stok tergantung pada *selling rate*. *Selling rate* diasumsikan sebagai fungsi pada persediaan saat ini dan *deterioration rate* adalah konstan. *Replenishment time* diasumsikan *zero lead time* dan model yang dikembangkan menggabungkan beberapa aspek *backlogging*, yaitu: *complete*, *partial*, dan *no backlogging*. Pada penelitian ini, EOQ ditentukan untuk memaksimalkan *total* keuntungan di setiap situasi. Sementara itu, Wu et al. (2006) melakukan penelitian untuk menentukan kebijakan *replenishment* optimal untuk *non-instantaneous perishable item* dimana permintaan tergantung pada stok. Pada penelitian ini, model persediaan dikembangkan dengan mengizinkan *shortages* dan *backlogging rate* yang merupakan variabel dan tergantung pada waktu tunggu *replenishment* selanjutnya. Penelitian ini membahas beberapa kasus, yaitu: model dengan *instantaneous* dan *non-instantaneous deterioration rate* serta model dengan atau tanpa *shortages*.

2.1.2 Review Pricing Policies pada Produk Perishable

Penelitian kebijakan persediaan produk *perishable* untuk merespon dinamika permintaan konsumen umumnya berfokus pada beberapa *variable reponse*, seperti *replenishment policy* dan *lot-size*. Namun, berdasarkan penelitian lainnya *pricing* menjadi *variable response* yang dapat dipertimbangkan sehingga ketiga *variable response* tersebut dapat dikombinasikan untuk mengembangkan model persediaan yang bertujuan memaksimalkan keuntungan. Contohnya, penelitian yang dilakukan oleh Taylor et al. (2014) mengembangkan kebijakan penelitian dibidang produk *perishable* dengan memperhatikan *replenishment*

policy dan permintaan yang dipengaruhi oleh *stock quantity display* serta harga jual produk. Tujuan dari penelitian ini adalah untuk memaksimalkan keuntungan yang diperoleh dengan menentukan kombinasi *variable* di atas menggunakan prinsip “*pontryagin*” *maximum*. (Abad, 1996) memformulasikan model *dynamic pricing* dan *lot-sizing* pada *reseller* yang menjual produk *perishable*. Abad mengasumsikan aktivitas *backorder* dapat dilakukan selama hal tersebut menguntungkan. Ketika produk merupakan *highly perishable*, *reseller* perlu melakukan *backlog* pada permintaan agar produk dipasarkan dengan harga yang masuk akal. Penelitian ini menunjukkan prosedur solusi sederhana untuk memecahkan masalah optimasi. Prosedur memerlukan pemecahan persamaan *single non-linear* dan jika dibutuhkan memecahkan dua persamaan *non-linear*.

(Mishra, 2017) mengembangkan model mengenai kebijakan yang optimal untuk produk *perishable* ketika permintaan tergantung pada kesegaran *displayed stock* dan harga jual. Pada penelitian tersebut, *displayed stock* memiliki peran yang efektif untuk meningkatkan permintaan karena mendorong pelanggan untuk membeli lebih banyak. Pada produk *perishable*, kesegaran (*freshness*) merupakan faktor yang mempengaruhi permintaan karena pelanggan lebih menyukai *fresh goods*. Selain itu, harga jual juga merupakan faktor yang mempengaruhi permintaan. Model persediaan ini mengembangkan asumsi dari model tradisional, yaitu persediaan akhir yang bernilai 0 dikembangkan menjadi *non-zero ending inventory*. Hal ini dilakukan karena menganggap produk yang diobral dengan harga lebih murah akan memberikan keuntungan dan selalu menjaga *on-hand fresh displayed stock* jika permintaan tergantung pada kesegaran dan stok. (Wee, 1999) menentukan model persediaan dengan asumsi skema *quantity discount* untuk biaya per unit, *partial backlogging* merupakan *fixed rate*, stok mengalami deteriorasi dalam waktu, dan *demand rate* merupakan fungsi linier dari harga jual. (Papachristos & Skouri, 2003) mengembangkan model (Wee, 1999) dengan asumsi *demand rate* adalah fungsi menurun *convex* dari harga jual dan *backlogging rate* tergantung pada fungsi waktu. Hal ini memastikan bahwa *rate* dari *backlogged demand* meningkat seperti waktu tunggu sampai waktu *replenishment* mengalami penurunan.

Penelitian lain yang membahas *pricing* sekaligus persediaan dilakukan oleh (Valliathal & Uthayakumar, 2011) dengan mengembangkan model EOQ untuk *non-instantaneous deteriorating item* dan mengizinkan adanya *shortages*. Penelitian ini dilakukan pada produk *perishable* seperti makanan, obat-obatan, dan kimia dengan tujuan untuk mengoptimalkan penentuan harga dan kebijakan *replenishment*. Penelitian ini mempertimbangkan *single product* pada ritel di suatu *infinite horizon* waktu dengan permintaan tergantung pada waktu dan harga. *Shortages* pada penelitian ini menggunakan *partial backlogging rate* dengan *infinite time horizon*. Namun, *shortages* pada model penelitian ini dimulai dengan tidak adanya *shortages* pada kebijakan *replenishment*. *Backlogging rate* adalah *non-increasing* fungsi waktu tunggu sampai waktu *replenishment* selanjutnya. Sedangkan permintaan adakah fungsi linier terhadap waktu dan menggunakan *partial backlogging* pada fungsi menurun dari waktu tunggu sampai waktu *replenishment* selanjutnya. Fungsi objektif dari model ini adalah memaksimalkan keuntungan dengan variabel keputusan harga jual, kuantitas *replenishment* (EOQ), waktu siklus *replenishment*, dan level persediaan yang bernilai positif. Valliathal dan Uthayakumar (2011) membandingkan keuntungan pada model *backlogging* eksponensial dan model *backlogging time-proportional*. Selain itu, (L. Feng et al., 2017) juga mengembangkan model *pricing* sekaligus persediaan. Penelitian tersebut mengembangkan kebijakan penetapan harga produk *perishable* dan jumlah *lot-size* dengan mempertimbangkan *expiration date*, *display stock*, dan *selling-price*. Metode yang digunakan untuk menyelesaikan permasalahan tersebut menggunakan pendekatan model matematika. Penelitian tersebut dilakukan untuk membuktikan beberapa fakta, yaitu: 1.) permintaan produk *perishable* tergantung pada harga, kesegaran, dan *stock level*; 2.) usia produk tidak hanya mengurangi *on hand stock*, tetapi juga menurunkan tingkat permintaan; 3.) produk *perishable* tidak dapat dijual setelah *expiration date* atau *maximum lifetime*. Berdasarkan teori pemasaran dan ekonomi, harga merupakan faktor utama yang mempengaruhi permintaan. Pada produk *perishable*, permintaan tergantung pada *freshness* dari produk itu sendiri sehingga *freshness index* tergantung pada usia produk. Wu et al. (2006) mengasumsikan *freshness* pada waktu t akan menurun secara linear dari 1 sampai 0 pada akhir siklus hidup.

Selain itu, permintaan juga meningkatkan *displayed stock* pada *fresh goods*, seperti: buah-buahan, sayur-sayuran, roti, susu, daging, dan makanan laut dapat lebih mendorong keinginan pelanggan untuk membeli sehingga formula untuk mencari *demand rate* adalah sebagai berikut:

$$D(t, P) = \alpha w^\beta \frac{m-t}{m} e^{-\lambda P} ; 0 \leq t \leq t_1 \leq T \quad (2.2)$$

Dimana α, β , dan λ adalah parameter tetap. Persamaan 2.2 dibatasi dengan $\alpha > 0$, $0 \leq \beta < 1$, dan $\lambda > 0$. Pada formula tersebut, α adalah jumlah maksimum dari *potential customer*, β adalah sensitivitas ruang rak (*shelf space*) terhadap *demand*, λ adalah sensitivitas harga terhadap *demand*. Artinya, *potential customer* tidak boleh negatif, sensitivitas ruang rak adalah antara 0 dan 1, serta sensitivitas harga harus positif, sehingga semakin besar harga permintaan akan menurun karena rumus di atas menunjukkan fungsi *exponential decay*. Selain itu, w adalah ukuran ruang rak dalam unit, m adalah usia maksimal produk, t adalah periode waktu, dan P adalah harga per unit. Selain itu, periode pada formula tersebut dibatasi yang dimulai dari periode 0 sampai periode T (*replanishment cycle time*). Sedangkan formula untuk mencari hasil optimal pada penelitian adalah:

$$\begin{aligned} \text{Max } TP(E, P, T) = & \frac{1}{T} \left\{ P(Q - E) + sE - cQ - o - h \left[\alpha w^\beta e^{-\lambda P} \left(\frac{t_1^3}{6m} - \frac{t_1^2}{2} \right) + Qt_1 \right] - \frac{h}{2} (w + \right. \\ & \left. E)(T - t_1) \right\} \end{aligned} \quad (2.3)$$

Subject to:

$$Q = w + \frac{\alpha w^\beta e^{-\lambda P}}{2m} \left[m^2 - (m - T)^2 - \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)e^{-\lambda P}} \right] \geq w \quad (2.4)$$

$$t_1 = m - \sqrt{(m - T)^2 + \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)e^{-\lambda P}}} \geq 0 \quad (2.5)$$

$$0 \leq E \leq w, \text{ dan } 0 \leq t_1 \leq T \leq m \quad (2.6)$$

Dimana:

T : *replanishment cycle time* dalam unit waktu

Q : *order quantity* dalam unit

E : persediaan akhir dalam unit ; $E \geq 0$

s : harga sisa (*salvage*) per unit

- o : biaya pesan
- h : biaya penyimpanan
- c : harga beli
- t_1 : waktu ketika *inventory level* mencapai w

Dimana fungsi objektif tersebut dibatasi oleh Q (*order quantity*) harus lebih besar atau sama dengan w (ukuran ruang rak) karena produk yang ada di rak *display* harus selalu diisi ulang ketika produk terjual atau dapat dikatakan rak *display* tidak boleh kosong sampai tidak ada lagi stok di gudang pada periode t_1 (waktu ketika *inventory level* mencapai w). Sehingga, t_1 harus lebih besar atau sama dengan 0 karena siklus hidup produk dimulai dari periode 0 . Selain itu, persediaan akhir harus lebih besar atau sama dengan 0 karena diasumsikan persediaan adalah *non-zero inventory*. Selain itu, Datta dan Pal (1990) mengeksplor bahwa permintaan tergantung pada stok karena semakin banyak persediaan maka semakin banyak permintaan. Penelitian tersebut membuktikan bahwa menjaga *level displayed stock* agar tetap terpenuhi akan menghasilkan keuntungan yang besar. Kebanyakan dari produk *perishable* hanya memiliki beberapa hari usia hidup dan *rate* permintaan pelanggan akan menurun sampai 0 ketika mendekati waktu kadaluarsa sehingga c mengembangkan waktu siklus *replenishment* yang optimal dan level persediaan akhir pada ritel ketika *rate* permintaan merupakan fungsi *multivariate* dari *freshness* produk dan level *displayed stock*.

Penelitian-penelitian di atas merupakan penelitian yang pernah dilakukan mengenai harga pada kebijakan persediaan produk *perishable*. Lebih jelas untuk perbandingan penelitian tersebut dapat dilihat pada.

2.1.3 Multi Item pada Manajemen Persediaan dan Harga Produk Perishable

Penelitian terkait kebijakan penentuan harga dan persediaan pada produk *perishable* pada umumnya dilakukan untuk *single item*. Pengembangan model tersebut akan lebih kompleks dan menarik jika dilakukan untuk *multi item*. Namun, baru sedikit penelitian yang menggabungkan model penentuan harga dan kebijakan persediaan pada produk *perishable*, contohnya penelitian yang dilakukan oleh Raouf (1993) membahas tentang masalah persediaan pada *multi*

item dan *single period*. *Single Period Inventory Problem* (SPIP) biasanya digunakan pada saat membuat *single order quantities* untuk mengantar produk sebelum dimulainya suatu periode dalam beberapa *planning horizon*. Pada SPIP diasumsikan bahwa pemesanan selanjutnya di dalam periode tidak layak dilakukan secara ekonomis. Hal ini terjadi ketika sebuah perusahaan berurusan dengan komoditas *perishable* yang memiliki usia penyimpanan terbatas. Biasanya SPIP ditujukan untuk *single item* dengan permintaan terdistribusi *uniform*. Namun, penelitian yang dilakukan oleh Daya dan Raouf mempertimbangkan *multi item* SPIP dengan konstrain modal dan *shelf space* (ruang penyimpanan/rak). Penelitian lain mengenai *multi item* pada model persediaan dilakukan oleh Taleizadeh et al. (2010). Penelitian tersebut dilakukan dengan menggunakan Algoritma Genetika untuk mengoptimalkan sistem persediaan pada *multi item* dan *multi constraint* dengan interval *replenishment* stokastik dan mengizinkan adanya diskon.

Penelitian mengenai *multi item* lebih lanjut dikembangkan oleh Shavandi et al. (2012) pada produk *perishable* dengan penentuan harga dan persediaan. Penelitian ini membagi tiga kategori produk, yaitu: independen, substitusi, dan pelengkap. Dari pengembangan hubungan dan dampak harga ketiga kategori produk tersebut ditemukan model permintaan umum. Fungsi permintaan pada ketiga jenis produk di atas dimodelkan sebagai berikut:

- a. Produk independen, yaitu produk dengan permintaan yang tidak mempertimbangkan produk lain dapat diformulasikan sebagai berikut:

$$D_i = m_i e^{-\alpha_i P_i}; i = 1, 2, \dots, n \quad (2.7)$$

- b. Produk substitusi, yaitu produk i yang memiliki produk pengganti j dapat diformulasikan sebagai berikut:

$$D_i = m_i e^{-\alpha_i P_i + n_j P_j}; j = 1, 2, \dots, n \quad (2.8)$$

- c. Produk pelengkap, yaitu produk i yang memiliki produk pelengkap k dapat diformulasikan sebagai berikut:

$$D_i = m_i e^{-\alpha_i P_i - \theta_k P_j}; k = 1, 2, \dots, n \quad (2.9)$$

Dengan batasan harga yang mempertimbangkan produk lain, yaitu:

$$\mu_i - \alpha_i \sigma_i' \leq P_i \leq \mu_i + \alpha_i \sigma_i'; k = 1, 2, \dots, n \quad (2.10)$$

Pada batasan di atas μ_i merupakan harga rata-rata produk i , σ_i' merupakan standar deviasi harga produk i , dan a_i merupakan koefisien dengan $a_i \geq 1$. Selain itu, pada persamaan di atas, m adalah maksimum permintaan yang tersedia, α adalah sensitivitas harga produk i , P adalah harga jual, n adalah sensitivitas harga produk j , dan θ adalah sensitivitas harga produk k sehingga fungsi objektif untuk memaksimalkan keuntungan *multi item* adalah:

$$Z = \frac{1}{T} \sum_{i=1}^n (R_i - SC_i - PC_i - HC_i - BC_i) \quad (2.11)$$

Dimana T adalah *planning horizon*, R_i adalah *revenue*, SC_i adalah biaya *set up*, PC_i adalah biaya produksi, HC_i adalah biaya penyimpanan, dan BC_i adalah biaya *shortage*.

Tujuan dari model yang dikembangkan oleh Shavandi et al. (2012) adalah memaksimalkan keuntungan dengan menentukan harga, persediaan, dan produksi. Model yang dikembangkan merupakan model *non-linear programming* dan diselesaikan dengan Algoritma Genetika. Untuk meningkatkan kinerja Algoritma Genetika, Shavandi et al (2012). Menggunakan metode *Taguchi Experimental Design* untuk menyesuaikan parameter algoritma.

Selain itu, berdasarkan (Gilarso, 2003) keterkaitan permintaan dengan harga dibedakan berdasarkan 3 kategori produk, yaitu:

- a. Produk independen, yaitu produk lepas yang tidak memiliki keterkaitan dengan produk manapun. Jika harga produk i naik ataupun turun maka tidak akan mempengaruhi harga produk lain atau produk j . Namun, mungkin saja akan mempengaruhi pendapatan riil karena mempengaruhi jumlah barang/jasa yang diminta.
- b. Produk substitusi (produk pengganti), yaitu produk yang memiliki hubungan timbal balik akibat kenaikan ataupun penurunan harga dari suatu produk yang memiliki pengganti. Seperti kereta api dengan bus dan kopi dengan the. Jika harga tiket kereta api naik maka orang akan lebih banyak yang membeli tiket bus. Jadi, permintaan akan bus sebagai kendaraan pengganti kereta akan lebih banyak akibat kenaikan harga dari tiket kereta, begitupun sebaliknya. Jika harga tiket kereta api turun maka permintaan akan kereta api sebagai kendaraan akan lebih banyak dan permintaan akan bus akan

menurun akibat penurunan harga tiket kereta. Atau dapat dikatakan perbandingannya adalah berbanding terbalik.

- c. Produk komplementer (pelengkap), yaitu produk yang saling melengkapi satu sama lain berdasarkan fungsinya. Dalam hal ini, jika harga produk i naik maka produk j sebagai pelengkapanya juga akan naik, begitupun sebaliknya. Jika harga produk i turun maka harga produk j sebagai pelengkapanya juga akan turun. Beberapa contoh produk komplementer antara lain: teh dengan gula, pulpen dengan tinta, dan kendaraan dengan bensin.

Penelitian-penelitian di atas merupakan penelitian yang pernah dilakukan mengenai *multi item* pada manajemen persediaan dan harga produk *perishable*.

2.1.4 Review Manajemen Persediaan dan Penetapan Harga dengan Batasan Kapasitas Gudang

Manajemen persediaan dan kebijakan penetapan harga merupakan 2 hal yang dapat mengoptimalkan *profit*. Namun, pada kenyataannya *retailer* atau pembeli memiliki kapasitas gudang yang terbatas. Sub bab ini membahas penelitian terdahulu mengenai manajemen persediaan dan kebijakan penetapan harga dengan batasan kapasitas gudang.

Beberapa penelitian lain mengenai manajemen persediaan mempertimbangkan kapasitas gudang sebagai batasan dalam menentukan nilai *order quantity* dari pembeli serta kapasitas produksi manufaktur. Wang et al. (2012) mempertimbangkan *single-manufacturer* dan *single-buyer* dalam rantai pasok dengan *single-deteriorating product* kebijakan konsinyasi. Penelitian ini juga menggunakan model *inventory control* yang terintegrasi dalam menentukan *batch* produksi dan *replenishment lot sizes*. Kemudian, pada penelitian ini dilakukan skenario dengan menggunakan batasan kapasitas gudang dan tanpa kapasitas gudang. Selain itu, Sana (2013) juga mempertimbangkan kapasitas gudang sebagai batasan dalam model *economic order quantity* dengan ketidakpastian permintaan. Penelitian ini juga menggunakan 2 tipe gudang, yaitu gudang milik sendiri dan sewa gudang jika dibutuhkan. Penelitian ini memformulasikan fungsi biaya rata-rata untuk distribusi *continuous* dan *discrete*

dari fungsi permintaan dengan membandingkan *holding cost* dan *stock out penalty*. Feng et al. (2017) meneliti tentang model persediaan yang terintegrasi dengan kapasitas gudang. Model ini terdiri dari *single-vendor* dan *multiple-buyer* dengan *warehouse capacity sharing* via *transshipment*. Penelitian ini mengoptimisasi kebijakan *transshipment* dengan mengembangkan model *nonlinear programming* dan algoritma genetika. Kebijakan persediaan ini dapat mengurangi channel-wide dan biaya tetap *transshipment* yang secara signifikan mempengaruhi kinerja algoritma. Dalam penelitian tersebut, analisis sensitivitas menunjukkan bahwa parameter memiliki efek yang berbeda terhadap biaya *channel-wide* dan kinerja algoritma. Selain itu, Tiwari et al. (2018) melakukan penelitian terhadap kebijakan penetapan harga dan *lot-sizing* untuk sistem rantai pasok pada *deteriorating item* dengan batasan kapasitas gudang. Penelitian ini memiliki pendekatan harga produk dan integrasi antara retailer dan pemasok berdasarkan 4 kebijakan, yaitu: terintegrasi, tidak terintegrasi, kebijakan *supplier-led Stackelberg*, dan *retailer-led Stackelberg policy*. Dalam penelitian ini, menganalisis efek harga, tingkat persediaan, dan integrasi antara *retailer* dan pemasok dengan skenario pada ritel yang memiliki batasan ruang rak. Bylka (2019) meneliti tentang *stock in stock* dengan batasan kapasitas gudang pembeli. Penelitian ini mengembangkan model deterministik *sustainable* dengan proses *centralized decision-making*. Rangkuman penelitian pada sub bab ini dapat dilihat pada tabel 2.2.

Tabel 2. 2 Rangkuman Manajemen Persediaan dengan Batasan Kapasitas Gudang

No.	Author	Year	Perishable Item	Fresh Good	Decision Variable		Warehouse Capacity
					Inventory	Pricing	
1	Wang et al.	2012	√		√		√
2	Sana	2013			√		√
3	Feng et al.	2017			√		√
4	Tiwari et al.	2018	√		√	√	√
5	Bylka	2019			√		√

2.2 *Artificial Bee Colony*

Artificial Bee Colony (ABC) adalah algoritma metaheuristik berbasis kawanan (*swarm*) yang diperkenalkan oleh Karaboga pada tahun 2005 untuk mengoptimalkan masalah numerik. Algoritma tersebut terinspirasi dari perilaku mencari makan lebah madu. Algoritma ini secara khusus didasari pada model yang diusulkan oleh Tereshko dan Loengarov (2005) untuk perilaku mencari makan koloni lebah madu. Model ini terdiri dari tiga komponen penting, yaitu: *employed bee*, *unemployed bee*, dan sumber makanan (*food sources*) (Karaboga & Basturk, 2008).

2.2.1 Perilaku Koloni Lebah Madu

Model minimum dalam koloni lebah madu terdiri dari 3 (tiga) komponen, yaitu: sumber makanan (*food sources*), pencari makanan (*employed foragers*), dan pencari makanan yang tidak bekerja (*unemployed foragers*). Sedangkan jenis perilakunya, yaitu: menghampiri sumber nektar dan meninggalkan sumber nektar (Karaboga & Basturk, 2008). Berikut adalah penjelasan mengenai komponen model:

a. *Food Sources*

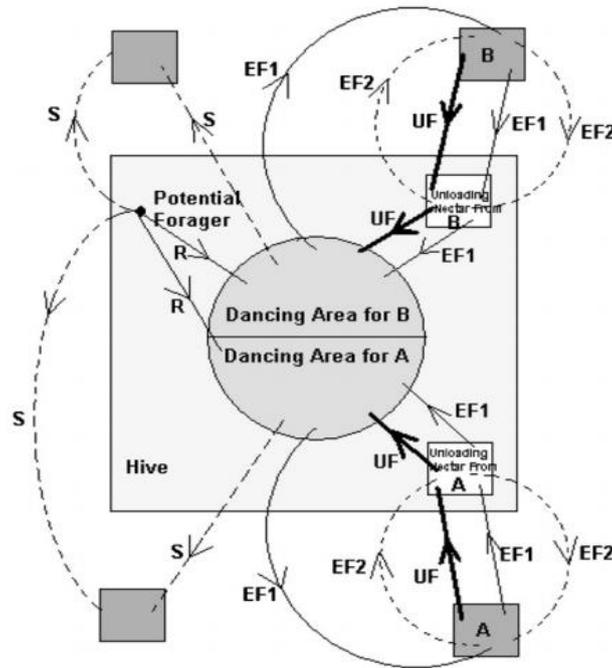
Nilai dari *food source* tergantung pada banyak faktor, seperti: jarak dengan sarang, kandungan energi, dan kemudahan mengekstraksi energi. Dengan tujuan untuk menyederhanakan, “*profitabilitas*” sumber makanan dapat diwakili dengan satu kuantitas.

b. *Employed Foragers*

Employed foragers membawa informasi tentang *food source* mengenai jarak dan arah dari sarang, serta peluang *profitabilitas* dari *food source* tersebut.

c. *Unemployed Foragers*

Unemployed foragers bertugas mencari *food source* untuk dieksploitasi. Terdapat dua tipe *unemployed foragers*, yaitu: 1.) *scouts*, bertugas mencari *food sources* baru di sekitar sarang, dan 2.) *onlookers*, bertugas menunggu di sarang dan membangun *food source* melalui informasi yang diperoleh dari *employed foragers*.



Gambar 2. 1 Perilaku lebah madu dalam mencari nektar

Gambar 2.1 menjelaskan dasar karakteristik perilaku *forager*. Asumsikan terdapat dua *food source*, yaitu: A dan B. Dimulai dengan lebah sebagai calon *forager* yang berperan sebagai *unemployed forager*. Lebah tersebut tidak memiliki pengetahuan mengenai sumber makanan di sekitar sarang.

Menurut (Karaboga & Basturk, 2008) terdapat dua pilihan yang memungkinkan untuk lebah tersebut, yaitu:

1. Lebah dapat menjadi *scout* (pengintai) dan mulai mencari makanan di sekitar sarang secara spontan (S dalam gambar 2.1).
2. Lebah dapat merekrut setelah melihat tarian dan mulai mencari sumber makanan (R dalam gambar 2.1).

Setelah menemukan sumber makanan, lebah menggunakan kapasitasnya untuk mengingat lokasi dan langsung mengeksploitasi sumber makanan. Oleh karena itu, lebah tersebut menjadi *employed foreger*. Kemudian, lebah tersebut mengambil nektar dan kembali ke sarang untuk meletakkan makanan ke tempat penyimpanannya. Setelah meletakkan makanan, lebah memiliki beberapa pilihan, yaitu:

1. Menjadi pengikut yang tidak berkomitmen (*uncommitted follower*) setelah meninggalkan sumber makanan. (UF)

2. Menari dan kemudian merekrut pasangan sebelum kembali ke sumber makanan yang sama. (EF1)
3. Terus mencari makanan di sumber makanan tanpa merekrut lebah lain. (EF2)

Perlu diketahui bahwa tidak semua lebah mulai mencari makan secara serentak. Berdasarkan percobaan, lebah baru akan mulai mencari makan dengan proporsi yang sebanding antara *total* lebah dan jumlah *forager bee* saat ini.

2.2.2 Algoritma Artificial Bee Colony

Koloni dalam *artificial bee* pada *Algoritma Artificial Bee Colony* terdiri dari tiga kelompok lebah, yaitu: *employed bee*, *onlooker*, dan *scout*. Kelompok pertama berisi *employed artificial bees* dengan jumlah setengah dari koloni dan kelompok kedua berisi *onlookers* dengan jumlah sisanya. Untuk setiap sumber makanan, hanya diperbolehkan satu *employed bee*. Dengan kata lain, jumlah *employed bee* sama dengan jumlah sumber makanan. Sedangkan *employed bee* yang meninggalkan sumber makanan menjadi *scout* (Karaboga & Basturk, 2008). Pencarian sumber makanan yang dilakukan oleh *artificial bees* dirangkum sebagai berikut:

- a. *Employed bee* menentukan sumber makanan di dalam lingkungan sumber makanan dalam ingatan mereka.
- b. *Employed bee* berbagi informasi dengan *onlookers* di dalam sarang lebah, kemudian *onlookers* memilih satu dari sumber makanan.
- c. *Onlookers* memilih sumber makanan di dalam lingkungan sumber makanan yang mereka pilih.
- d. *Employed bee* yang meninggalkan sumber makanannya menjadi *scout* dan mulai mencari sumber makanan baru secara acak.

Tahapan dalam *Algoritma Artificial Bee Colony*:

- *Initialize*
Generate random number
- *Repeat*:

1. Pindahkan *employed bee* ke sumber makanan mereka dan tentukan jumlah nektar mereka.
2. Pindahkan *onlooker* ke sumber makanan dan tentukan jumlah nektar mereka.
3. Pindahkan *scout* untuk mencari sumber makanan baru.
4. Simpan sumber makanan terbaik yang ditemukan.

- *Until*

Ketika solusi optimal telah didapatkan maka berhenti.

Setiap siklus pencarian terdiri dari tiga langkah, yaitu: memindahkan *employed* dan *onlooker bee* ke sumber makanan dan hitung jumlah nektar dan tentukan *scout bees*. Kemudian, pindahkan mereka secara acak ke sumber makanan yang memungkinkan. Sumber makanan merupakan solusi optimal yang memungkinkan untuk masalah tersebut. Jumlah nektar dari sumber makanan menggambarkan kualitas solusi yang diwakili oleh sumber makanan tersebut. Penempatan *onlooker* pada makanan dilakukan dengan menggunakan metode *roulette wheel selection*. Setiap koloni lebah memiliki *scout* yang merupakan penjelajah. Para penjelajah tidak memiliki panduan apa pun saat mencari makanan. Fokus utama mereka hanyalah menemukan segala jenis sumber makanan. Sebagai akibat dari perilaku tersebut, *scout* ditandai dengan biaya pencarian rendah dan rata-rata kualitas sumber makanan rendah. Sesekali, *scout* secara tidak sengaja dapat menemukan sumber makanan yang kaya dan sama sekali belum diketahui. Dalam kasus *artificial bee*, *artificial scout* dapat menemukan solusi *feasible* dengan cepat. Dalam algoritma ABC, salah satu *employed bee* dipilih dan diklasifikasikan sebagai *scout bee*. Klasifikasi dikendalikan oleh parameter yang disebut “batas”. Jika solusi yang mewakili sumber makanan tidak digantikan dengan jumlah percobaan yang telah ditentukan maka sumber makanan tersebut ditinggalkan oleh *employed bee* dan *employed bee* menjadi *scout*. Jumlah percobaan untuk meninggalkan sumber makanan sama dengan nilai “batas”, yang merupakan parameter kontrol penting dalam algoritma ABC.

Dalam proses pencarian yang kuat, proses eksplorasi dan eksploitasi harus dilakukan secara bersamaan. Dalam algoritma ABC, ketika *onlooker* dan

employed bees melakukan proses eksploitasi di ruang pencarian, *scout* mengontrol proses eksplorasi sehingga algoritma dapat dikatakan memiliki proses pencarian yang kuat. Dalam kasus nyata lebah madu, rekrutmen menggambarkan “ukuran” tentang seberapa cepat koloni lebah menemukan dan mengeksploitasi sumber makanan yang baru ditemukan. *Artificial recruiting* menggambarkan “pengukuran” dari kecepatan solusi yang layak atau solusi “*good quality*” dari masalah optimasi. Keberlangsungan hidup dari koloni lebah tergantung pada kecepatan menemukan dan memanfaatkan sumber makanan terbaik. Begitu pula dengan keberhasilan solusi, hal ini juga tergantung pada kecepatan menemukan “*good solution*” terutama untuk masalah yang membutuhkan penyelesaian secara *real time*.

Seperti *forager* lainnya, lebah mencari sumber makanan dengan cara memaksimalkan rasio E/T (di mana E adalah energi dan T adalah waktu yang dihabiskan untuk mencari sumber makanan). dalam kasus kumpulan lebah, E sebanding dengan jumlah nektar sumber makanan yang ditemukan oleh lebah dan kumpulan lebah bekerja untuk memaksimalkan madu yang disimpan di dalam sarang. Tujuan dalam masalah maksimalisasi adalah menemukan fungsi objektif maksimum $F(\theta)$, $\theta \in R^p$. Asumsikan bahwa θ_i menggambarkan jumlah nektar pada sumber makanan yang terletak di θ_i dan sebanding dengan energi $E(\theta_i)$. Misalkan $P(c) = \{\theta_i(c) | i = 1, 2, \dots, S\}$ (c : *cycle*, S : jumlah sumber makanan di sekitar sarang) menggambarkan populasi sumber makanan yang dikunjungi lebah.

Seperti disebutkan sebelumnya, preferensi sumber makanan oleh *onlooker bee* tergantung pada jumlah nektar $F(\theta)$ dari sumber makanan. Karena jumlah nektar dari sumber makanan meningkat, probabilitas sumber yang disukai oleh *onlooker bee* meningkat secara proporsional. Oleh karena itu, probabilitas sumber makanan yang akan dipilih dan terletak di (θ_i) dapat diformulasikan sebagai berikut:

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^S F(\theta_k)} \quad (2.12)$$

Setelah melihat tarian *employed bee*, *onlooker bee* pergi ke daerah sumber makanan yang berlokasi di (θ_i) . Oleh karena itu, probabilitas dan penentuan lingkungan sumber makanan tergantung pada beberapa informasi visual, seperti

tanda yang berada pada *patch*. Dengan kata lain, *onlooker bee* memilih satu dari banyak sumber makanan setelah melakukan perbandingan di antara sumber makanan sekitar (θ_i) sehingga posisi lingkungan sumber makanan dihitung dengan formula sebagai berikut:

$$\theta_i(c + 1) = \theta_i(c) \pm \phi_i(c) \quad (2.13)$$

$\phi_i(c)$ merupakan langkah yang diperoleh secara acak untuk menemukan sumber makanan dengan lebih banyak nektar di sekitar θ_i . $\phi_i(c)$ dihitung dengan mengambil selisih dari bagian yang sama pada $\phi_i(c)$ dan $\phi_k(c)$ (k merupakan indeks acak) di posisi makanan. Jika jumlah nektar $F(\theta_i(c + 1))$ pada $\theta_i(c + 1)$ lebih besar dibandingkan pada $\theta_i(c)$, maka lebah akan pergi ke sarang dan membagikan informasi kepada lebah lainnya dan posisi $\theta_i(c)$ pada sumber makanan berubah menjadi $\theta_i(c + 1)$, jika tidak $\theta_i(c)$ tetap seperti semula.

Setiap sumber makanan hanya memiliki satu *employed bee*. Oleh karena itu, jumlah *employed bee* sama dengan jumlah sumber makanan. Jika posisi θ_i pada sumber makanan i tidak dapat diperbaiki melalui jumlah percobaan “limit” yang telah ditentukan sebelumnya, maka sumber makanan θ_i ditinggalkan oleh *employed bee* dan *employed bee* menjadi *scout*. Kemudian, *scout* mulai mencari sumber makanan baru. Setelah itu, posisi baru dijadikan θ_i .

2.3 Review Metaheuristik pada Manajemen Persediaan

Salah satu metode yang dapat digunakan dalam memecahkan masalah adalah metaheuristik. Beberapa penelitian menggunakan metode metaheuristik untuk memecahkan masalah pada manajemen persediaan. Penelitian terkait manajemen persediaan dengan menggunakan metode metaheuristik dapat dilihat pada table 2.3.

Tabel 2. 3 Perbandingan penggunaan metode metaheuristik

Peneliti	Tujuan	Metaheuristik	Kinerja terbaik
Karaboga dan Basturk (2007)	Membandingkan kinerja algoritma ABC dengan <i>differential evolution</i> (DE), <i>particle swarm optimization</i> (PSO) dan <i>evolutionary</i>	<i>Differential Evolution</i> (DE), <i>Particle Swarm Optimization</i> (PSO), dan <i>Evolutionary Algorithm</i> (EA) atau GA, dan	<i>Algoritma Artificial Bee Colony</i> (ABC)

	<i>algorithm</i> (EA) untuk masalah numerik multi-dimensi	<i>Algoritma Artificial Bee Colony</i> (ABC)	
Talaizadeh et al., (2010)	Mengoptimalkan sistem persediaan pada <i>multi item</i> dan <i>multi constraint</i>	<i>Genetic Algorithm</i> (GA) dan <i>Simulated Annealing</i> (SA)	<i>Genetic Algorithm</i> (GA)
Shavandi et al., (2012)	Menyajikan model baru mengenai penetapan harga dan persediaan multi-produk	<i>Genetic Algorithm</i> (GA)	<i>Genetic Algorithm</i> (GA)
Bhunia et al. (2017)	Mengembangkan model persediaan pemutusan harga <i>deteriorating item</i>	<i>Particle Swarm Optimization</i> (PSO)	<i>Particle Swarm Optimization</i> (PSO)

Seperti yang telah dibahas sebelumnya, (Talaizadeh et al., 2010) telah melakukan penelitian mengenai manajemen persediaan dengan menggunakan algoritma genetika untuk mengoptimalkan sistem persediaan pada *multi item* dan *multi constraint* dengan *interval replenishment* stokastik dan mengizinkan adanya diskon. Pada penelitian tersebut, (Talaizadeh et al., 2010) menggunakan 2 (dua) metode metaheuristik yaitu *Genetic Algorithm* (GA) dan *Simulated Annealing* (SA) untuk menyelesaikan masalah pada penelitian tersebut. Hasilnya, metode GA menghasilkan nilai fungsi objektif yang lebih baik dibandingkan dengan SA. Selain itu, (Shavandi et al., 2012) juga telah melakukan penelitian pada masalah manajemen persediaan dan harga *multi product*. Penelitian ini menggunakan metode metaheuristik *Genetic Algorithm* (GA) dan lingo dalam menyelesaikan masalah tersebut. Perbandingan hasil keduanya menunjukkan bahwa kinerja GA lebih baik untuk fungsi objektif dibandingkan dengan lingo, yaitu 21,8% lebih besar dari lingo.

(Bhunia & Pareek, 2017) melakukan penelitian dengan menggunakan metode metaheuristik *Genetic Algorithm* (GA) dan *Particle Swarm Optimization* (PSO) pada model persediaan untuk *single deteriorating item* dengan variabel permintaan bergantung pada strategi pemasaran dan level *displayed stock*. Pada kehidupan nyata, *displayed stock* memerankan peran penting pada fungsi permintaan dan menarik konsumen untuk membeli produk lebih banyak. Selain itu, (Bhunia & Pareek, 2017) melakukan penelitian mengenai model harga dan persediaan pada *deteriorating item* dimana permintaan bergantung pada level *displayed stock* dan mengizinkan adanya *shortage partially backlogged* dengan

kebijakan diskon menggunakan metode *Particle Swarm Optimization* (PSO). Penelitian ini juga membandingkan penyelesaian masalah menggunakan *Particle Swarm Optimization* (PSO) dengan beberapa versi kuantum PSO yang berbeda, yaitu WQPSO, AQPSO, dan GQPAO dengan hasil kinerja sama.

Tabel 2. 4 Perbandingan hasil yang diperoleh dengan menggunakan metode DE, PSO, EA, dan algoritma ABC

	DE [26]	PSO [26]
$f_1(\vec{x})$	0 ± 0	0.00453 ± 0.00090
$f_2(\vec{x})$	0 ± 0	$2.51130E-8 \pm 0$
$f_3(\vec{x})$	0 ± 0	1.54900 ± 0.06695
$f_4(\vec{x})$	0 ± 0	13.1162 ± 1.44815
$f_5(\vec{x})$	35.3176 ± 0.27444	5142.45 ± 2929.47

	EA [26]	ABC
	0 ± 0	0 ± 0
	0 ± 0	0 ± 0
	0.00624 ± 0.00138	0 ± 0
	32.6679 ± 1.94017	0 ± 0
	79.8180 ± 10.4477	$0.133109389824 \pm 0.262242170275$

Mengenai metode metaheuristik, (Karaboga & Basturk, 2008) telah melakukan penelitian yang membandingkan kinerja *Algoritma Artificial Bee Colony* (ABC) dengan metode metaheuristik lainnya, yaitu *Differential Evolution* (DE), *Particle Swarm Optimization* (PSO) dan *Evolutionary Algorithm* (EA) atau *Genetics Algorithm* untuk masalah dengan *multi-dimensional numeric problems*. Hasil dari penelitian tersebut menunjukkan bahwa kinerja algoritma ABC dapat dibandingkan dengan metode di atas dan hasil simulasi juga menunjukkan bahwa algoritma ABC memiliki kinerja yang lebih baik daripada metode lainnya seperti disebutkan sebelumnya seperti pada tabel 2.4.

2.4 Gap Penelitian

Berdasarkan penelitian terdahulu belum ada yang mengembangkan model persediaan dan penentuan harga pada multi produk *perishable* khususnya *fresh*

goods di ritel dengan permintaan yang tergantung pada *expiration date*, *displayed stock*, dan *selling-price* dengan menggunakan *Algoritma Artificial Bee Colony* (ABC) sehingga penulis menjadikan *gap* tersebut sebagai topik pada penelitian ini. *Gap* penelitian pengembangan model persediaan dan penentuan harga dapat dilihat pada tabel 2.5 berikut.

Tabel 2. 5 Penelitian terdahulu dan *gap* penelitian

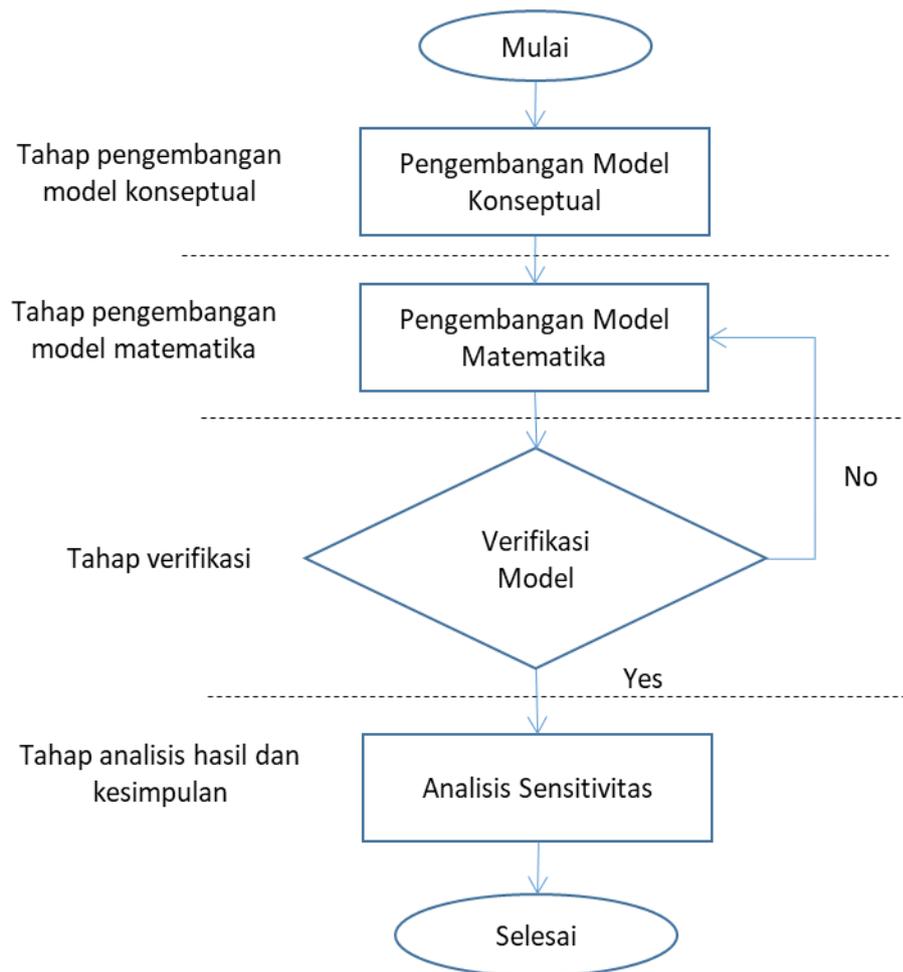
No.	Peneliti	Tahun	Perishable Item	Fresh Good	Decision Variable		Jenis Item		Parameter Permintaan			Kapasitas Gudang
					Inventory	Pricing	Single Item	Multi Item	Expiration Date	Displayed Stock	Selling Price	
1	Zoller	1977			√			√				√
2	Datta dan Pal	1990	√		√		√		√			
3	Padmanabhan dan Vrat	1993	√		√		√					
4	Daya dan Raouf	1993			√			√				
5	Wee	1999	√		√		√				√	
6	Chang dan Dye	1999	√		√		√					
7	Papachristos dan Skouri	2003	√		√		√				√	
8	Wu et al.	2005	√		√		√					
9	Dye	2007	√		√		√					√
10	Lee dan Wang	2008			√		√					√
11	Talaizadeh et al.	2010			√			√				
12	Valliathal dan Uthayakumar	2011	√		√	√	√				√	
13	Shavandi et al.	2012	√		√	√		√			√	
14	Wang	2012	√		√		√					√
15	Wang et al.	2012	√		√		√					√
16	Allahabad	2013	√	√	√		√			√		

No.	Peneliti	Tahun	Perishable Item	Fresh Good	Decision Variable		Jenis Item		Parameter Permintaan			Kapasitas Gudang
					Inventory	Pricing	Single Item	Multi Item	Expiration Date	Displayed Stock	Selling Price	
17	Sana	2013			√		√					√
18	Taylor et al.	2014	√		√		√			√	√	
19	Hidayat dan Fauzi	2015	√		√		√		√			
20	Abad	2015	√		√	√	√				√	
21	Feng et al.	2016	√	√	√	√	√		√	√	√	
22	Wu et al.	2016 a	√		√		√			√		
23	Dobson et al.	2017	√	√	√		√		√			
24	Mishra	2017	√	√	√		√			√	√	
25	Feng et al.	2017			√		√					√
26	Kirci et al.	2018	√		√		√					
27	Khurana	2018	√		√		√			√		
28	Tiwari et al.	2018	√		√	√	√			√	√	√
29	Bylka	2019			√		√					√
30	Penelitian Ini	2020	√	√	√	√		√	√	√	√	√

(Halaman ini sengaja dikosongkan)

BAB 3 METODOLOGI PENELITIAN

Bab ini menggambarkan pola pikir dan tahap-tahap dalam melakukan penelitian. Tahap-tahap tersebut terdiri dari 4 (empat) tahap, yaitu: tahap pengembangan model konseptual, tahap pengembangan model matematika, tahap verifikasi, dan tahap analisis hasil. Adapun tahap-tahap penelitian tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Flowchart* metodologi penelitian

3.1 Pengembangan Model Konseptual

Tahap pengembangan model konseptual yang menggambarkan kondisi sistem yang akan dimodelkan. Penelitian ini memodelkan *single item* produk *perishable* mengenai penetapan harga dan kebijakan persediaan untuk produk

perishable khususnya *fresh good* ketika permintaan tergantung pada *harga jual*, *displayed stock*, dan *expiration date*. Model tersebut dikembangkan menjadi model *multi item* produk *perishable* yang dikategorikan ke dalam 2 (dua) jenis, yaitu: produk independen dan produk substitusi. Kedua kategori produk ini dibedakan berdasarkan harga produk yang akan mempengaruhi *demand* (permintaan) produk lain. Peningkatan harga pada produk independen tidak memiliki efek terhadap produk lain. Namun, peningkatan harga pada produk substitusi akan mempengaruhi produk lainnya.

3.2 Pengembangan Model Matematika

Tahap ini menjelaskan mengenai proses pengembangan model matematika. Proses pengembangan model matematika pada penelitian ini dapat dilihat pada Gambar 3.2. Model matematika ini mengkombinasikan model matematika dari Feng et al., (2017) dan Shavandi et al., (2012). Model *single item* produk *perishable* dari Feng et al. (2017) dengan fungsi tujuan seperti persamaan 2.3:

$$\text{Max } TP(E, P, T) = \frac{1}{r} \left\{ P(Q - E) + sE - cQ - o - h \left[\alpha w^\beta e^{-\lambda P} \left(\frac{t_1^3}{6m} - \frac{t_1^2}{2} \right) + Q t_1 \right] - \frac{h}{2} (w + E)(T - t_1) \right\} \quad (2.3)$$

Dan fungsi kendala seperti persamaan 2.4, 2.5, dan 2.6, yaitu:

- a. Persamaan 2.4

$$Q = w + \frac{\alpha w^\beta e^{-\lambda P}}{2m} \left[m^2 - (m - T)^2 - \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)e^{-\lambda P}} \right] \geq w \quad (2.4)$$

- b. Persamaan 2.5

$$t_1 = m - \sqrt{(m - T)^2 + \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)e^{-\lambda P}}} \geq 0 \quad (2.5)$$

- c. Persamaan 2.6

$$0 \leq E \leq w, \text{ dan } 0 \leq t_1 \leq T \leq m \quad (2.6)$$

Dimana:

T : *replenishment cycle time* dalam unit waktu

Q: *order quantity* dalam unit

E: persediaan akhir dalam unit ; $E \geq 0$

P: harga per unit

s : harga sisa (*salvage*) per unit
 o : biaya pesan
 h : biaya penyimpanan
 c : harga beli
 t_1 : waktu ketika inventory level mencapai w
 α : jumlah maksimum dari *potential customer*
 β : sensitivitas ruang rak (*shelf space*) terhadap *demand*
 λ : sensitivitas harga terhadap *demand*
 w : ukuran ruang rak dalam unit
 m : usia maksimal produk
 t : periode waktu

Model di atas akan dikembangkan menjadi model *multi-item* produk *perishable* dengan mengadopsi fungsi permintaan yang telah dikembangkan oleh Shavandi et al. (2012) seperti persamaan 2.7, yaitu:

$$D_i = m_i e^{-\alpha_i P_i}; i = 1, 2, \dots, n \quad (2.7)$$

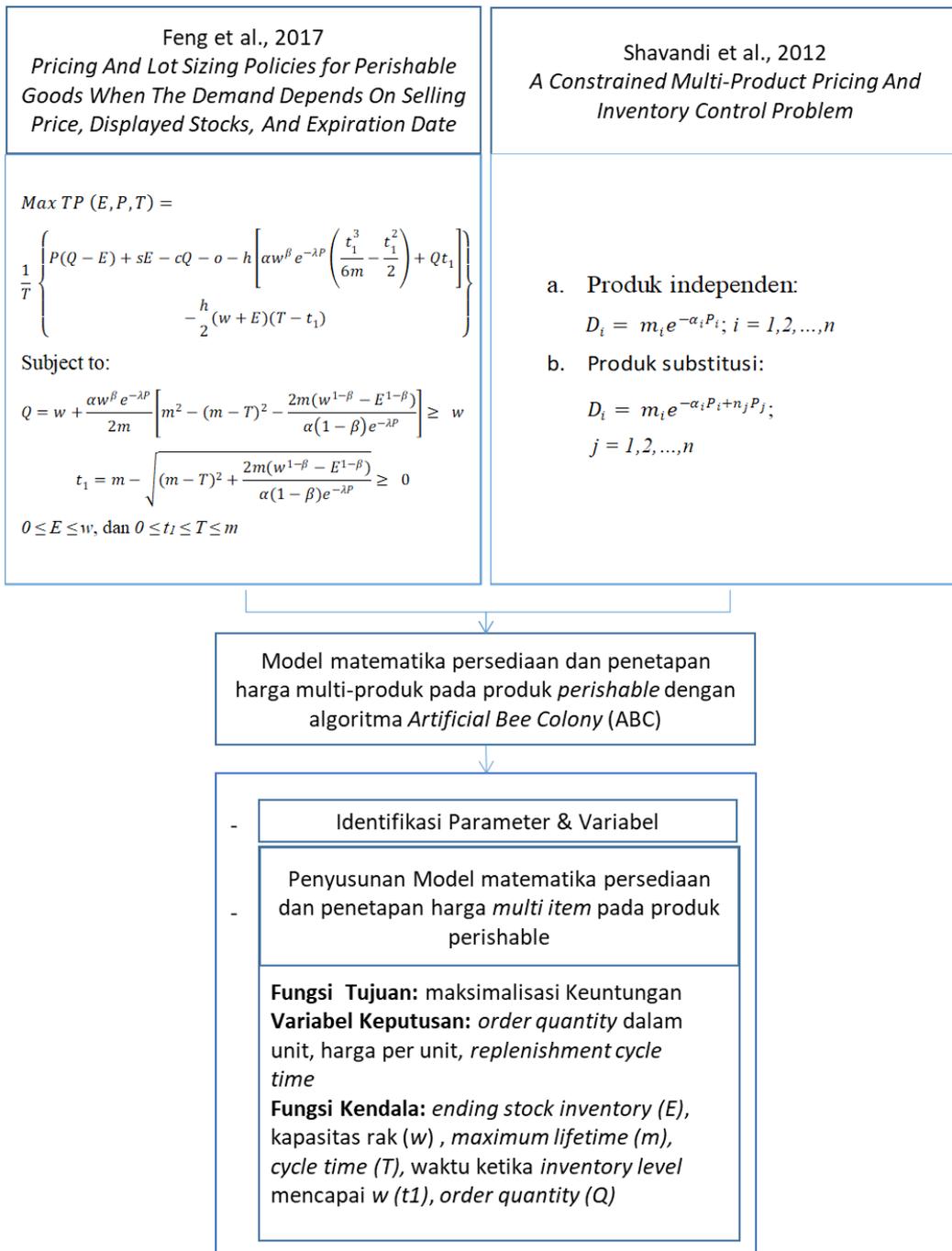
dan persamaan 2.8, yaitu:

$$D_i = m_i e^{-\alpha_i P_i + n_j P_j}; j = 1, 2, \dots, n \quad (2.8)$$

Dimana:

m : maksimum permintaan yang tersedia
 α : sensitivitas harga produk I
 P : harga jual
 n : sensitivitas harga produk j
 θ : sensitivitas harga produk k

Proses pengembangan model matematika ini dilakukan dengan menggunakan metode metaheuristik dengan *Algoritma Artificial Bee Colony* (ABC) dengan memasukan fungsi objektif, variabel keputusan, dan kendala.



Gambar 3.2 Tahapan pengembangan model

3.3 Verifikasi

Verifikasi merupakan proses yang digunakan untuk memastikan bahwa model yang dibuat sesuai dengan konsep yang telah ditetapkan sebelumnya. Proses verifikasi juga bertujuan untuk memastikan algoritma yang ditulis telah

sesuai dan memastikan model tidak memiliki eror. Proses ini dilakukan dengan *running* model matematika dengan contoh kasus kecil menggunakan *excel solver*. Pada penelitian ini, penulis juga melakukan verifikasi menggunakan metode metaheuristik dan membandingkan hasilnya dengan hasil awal dari data yang digunakan. Dalam hal ini, penulis menggunakan data dari jurnal acuan model penelitian ini. Sedangkan validasi adalah proses yang digunakan untuk menentukan apakah model telah sesuai dengan sistem nyata. Berdasarkan pengertian tersebut, pengujian pada penelitian ini cukup sampai proses verifikasi karena penelitian ini merupakan pengembangan model yang menerjemahkan model konseptual ke dalam bahasa pemrograman.

3.4 Analisis Hasil

Tahap analisis dimulai dengan membandingkan hasil percobaan numerik dengan beberapa skenario. Setelah itu, penulis melakukan analisis sensitivitas terhadap model matematika dengan untuk mengetahui pengaruh perubahan parameter terhadap nilai *profit* pada setiap kategori produk. Hasil tersebut diwujudkan dalam bentuk grafik sehingga analisis dan pembahasan dapat dilakukan. Kemudian, hasil dari analisis sensitivitas tersebut menjadi *output* untuk ditarik kesimpulan.

(Halaman ini sengaja dikosongka

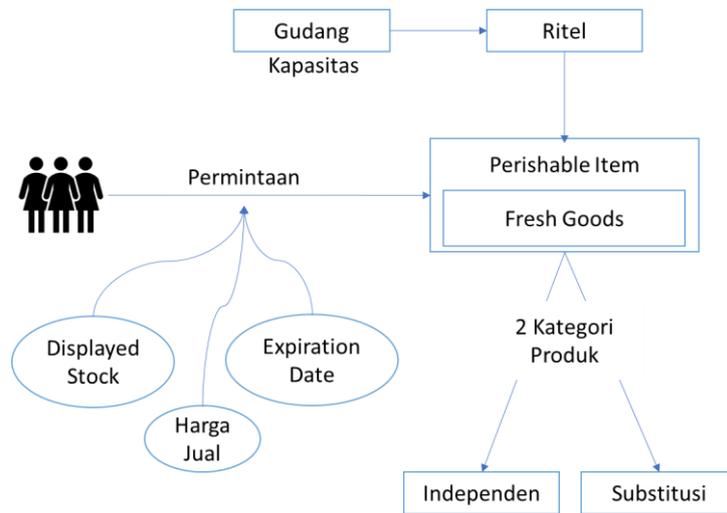
BAB 4

PENGEMBANGAN MODEL

Bab ini membahas proses dan struktur pengembangan model yang terdiri dari: pengembangan model konseptual dan pengembangan model matematika sesuai dengan tujuan pada penelitian ini.

4.1 Model Konseptual

Model yang dikembangkan dalam penelitian ini menggambarkan ritel yang menjual *fresh goods* berdasarkan model persediaan *single*-produk dari Feng et al., (2017) yang dikembangkan menjadi multi-produk. Model Feng et al., (2017) tersebut dikombinasikan dengan model Shavandi et al., (2012). Dimana permintaan pada model Feng et al., (2017) dipengaruhi oleh harga jual, *displayed stock*, dan *expiration date*. Sedangkan permintaan pada model Shavandi et al., (2012) hanya dipengaruhi oleh harga jual. Produk pada model Shavandi et al., (2012) terdiri dari tiga kategori, yaitu: produk independen, substitusi, dan komplementer. Namun, penelitian ini hanya mengadopsi dua kategori produk, yaitu: produk independen dan produk substitusi karena mempertimbangkan karakteristik dari *fresh goods* yang tidak masuk ke dalam kategori produk komplementer. Selain itu, penulis mengkombinasikan kapasitas gudang yang akan dibahas pada skenario lebih lanjut. Lebih jelasnya mengenai model konseptual ini dapat dilihat pada gambar 4.1



Gambar 4.1 Ilustrasi Konseptual Model

4.2 Model Matematika

Penyusunan model pada penelitian ini didasari dari model persediaan dan penetapan harga *single*-produk yang dikembangkan oleh Feng et al., (2017) dan fungsi permintaan dengan dua kategori produk yang mengacu pada Shavandi et al., (2012).

4.2.1. Model Dasar

Seperti yang telah disebutkan sebelumnya, model pada penelitian ini didasari dari penelitian yang dilakukan Feng et al., (2017) dan Shavandi et al., (2012). Berikut adalah penjabarannya:

- a. Model persediaan dan penetapan harga Feng et al., (2017)

Model dasar pada penelitian ini didasari oleh model persediaan dan penetapan harga yang mengacu pada Feng et al., (2017). Model tersebut adalah sebagai berikut:

- Fungsi objektif:

$$\begin{aligned}
 \text{Max } TP(E, P, T) = & \frac{1}{T} \left\{ P(Q - E) + sE - cQ - o - h \left[\alpha w^\beta e^{-\lambda P} \left(\frac{t_1^3}{6m} - \frac{t_1^2}{2} \right) + Qt_1 \right] - \right. \\
 & \left. \frac{h}{2}(w + E)(T - t_1) \right\} \quad (4.1)
 \end{aligned}$$

Formula *total profit* di atas terdiri dari *revenue* yang dikurangi oleh biaya-biaya terkait. *Revenue* pada formula di atas terdiri dari penjualan produk dengan harga normal dan penjualan produk dengan harga diskon atau *salvage*. Sedangkan komponen biaya pada formula di atas, terdiri dari: biaya pembelian, biaya pemesanan, dan biaya penyimpanan. Pada penelitian tersebut, *shortage* tidak diizinkan sehingga biaya *shortage* tidak masuk ke dalam komponen biaya.

- Batasan:

$$Q = w + \frac{\alpha w^\beta e^{-\lambda P}}{2m} \left[m^2 - (m - T)^2 - \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)e^{-\lambda P}} \right] \geq w \quad (4.2)$$

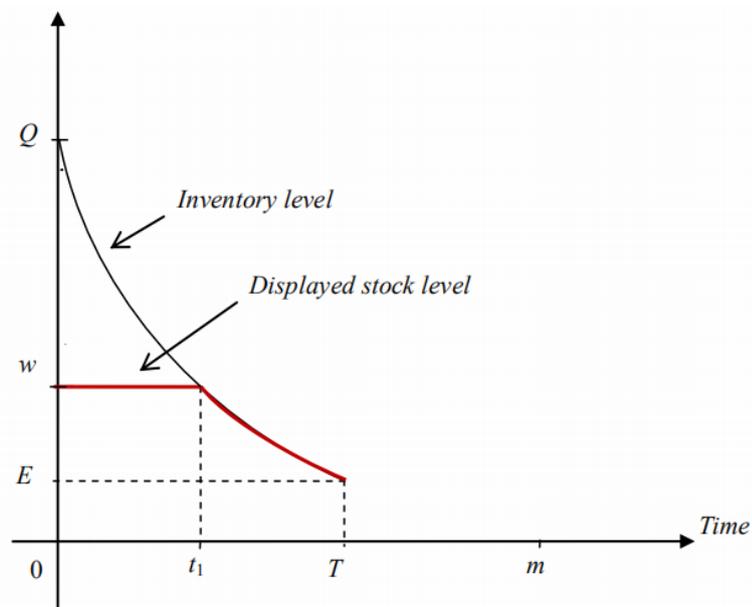
Batasan ini menjelaskan bahwa order *quantity* (Q) harus lebih besar atau sama dengan ukuran rak *display* (w). Hal ini bertujuan agar ruang rak dapat dimanfaatkan secara maksimal dan selalu terisi penuh sehingga pelanggan lebih tertarik untuk membeli produk. Selain itu, jika ritel tidak dapat memenuhi kuantitas pada rak *display* maka biaya penyimpanan pada rak *display* akan menjadi pemborosan.

$$t_1 = m - \sqrt{(m - T)^2 + \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)e^{-\lambda P}}} \geq 0 \quad (4.3)$$

Batasan ini menjelaskan bahwa waktu ketika *inventory level* mencapai kuantitas ukuran rak/tidak ada stok di gudang (t_1) harus lebih besar atau sama dengan 0. Hal ini dikarenakan siklus produk dimulai dari periode 0.

$$0 \leq E \leq w, \text{ dan } 0 \leq t_1 \leq T \leq m \quad (4.4)$$

Jika digambarkan melalui grafik, batasan ini dapat dilihat pada gambar 4.1.



Gambar 4.2 Grafik Sistem Persediaan Feng et al., (2017)

Pada model Feng et al., (2017) tersebut, fungsi permintaan dipengaruhi oleh harga jual, *displayed stock*, dan *expiration date* sehingga fungsi permintaan pada model tersebut adalah:

$$D(t, P) = \alpha w \beta^{\frac{m-t}{m}} e^{-\lambda P} ; 0 \leq t \leq t_1 \leq T \quad (4.5)$$

Dengan notasi variabel sebagai berikut:

T : *replanishment cycle time* dalam unit waktu

Q : *order quantity* dalam unit

E : persediaan akhir dalam unit ; $E \geq 0$

P : harga per unit

s : harga sisa (*salvage*) per unit

o : biaya pesan

h : biaya penyimpanan

c : harga beli

t_1 : waktu ketika inventory level mencapai w

α : jumlah maksimum dari *potential customer*

β : sensitivitas ruang rak (*shelf space*) terhadap *demand*

λ : sensitivitas harga terhadap *demand*

w : ukuran ruang rak dalam unit

m : usia maksimal produk

t : periode waktu

b. Fungsi permintaan Shavandi et al., (2012)

Model multi-produk berdasarkan Shavandi et al., (2012) dikategorikan ke dalam beberapa produk, diantaranya produk independen dan produk substitusi. Kedua kategori produk ini memiliki formula yang berbeda pada fungsi permintaan, yaitu:

- Produk independen, yaitu produk dengan permintaan yang tidak mempertimbangkan produk lain dapat diformulasikan sebagai berikut:

$$D_i = m_i e^{-\alpha_i P_i}; i = 1, 2, \dots, n \quad (4.6)$$

- Produk substitusi, yaitu produk i yang memiliki produk pengganti j dapat diformulasikan sebagai berikut:

$$D_i = m_i e^{-\alpha_i P_i + n_j P_j}; j = 1, 2, \dots, n \quad (4.7)$$

Dengan batasan harga yang mempertimbangkan produk lain, yaitu:

$$\mu_i - a_i \sigma_i' \leq P_i \leq \mu_i + a_i \sigma_i'; k = 1, 2, \dots, n \quad (4.8)$$

Pada batasan di atas, μ_i merupakan harga rata-rata produk i , σ_i' merupakan standar deviasi harga produk i , dan a_i merupakan koefisien dengan $a_i \geq 1$. Batasan ini digunakan karena mempertimbangkan situasi harga produk lain.

Dengan notasi variabel sebagai berikut:

m : maksimum permintaan yang tersedia

α : sensitivitas harga produk i

P : harga jual

n : sensitivitas harga produk j

c. Model multi-produk Shavandi et al., (2012)

Model multi-produk yang dikembangkan oleh penulis didasari oleh model Shavandi et al., (2012) dengan tujuan memaksimalkan produk. Sehingga, fungsi objektif pada model tersebut diformulasikan sebagai berikut:

$$Z = \frac{1}{T} \sum_{i=1}^n (R_i - SC_i - PC_i - HC_i - BC_i) \quad (4.9)$$

Fungsi di atas berisi komponen-komponen pembentuk *profit* yang dihasilkan dari penjualan produk. Komponen tersebut terdiri dari *revenue* yang dikurangi oleh biaya-biaya terkait pada model.

Dengan notasi variabel sebagai berikut:

T : *planning horizon*

R_i : *revenue*

SC_i : biaya *set up*

PC_i : biaya produksi

HC_i : biaya penyimpanan

BC_i : biaya *shortage*

Z : fungsi tujuan / fungsi objektif

4.2.2. Model Penelitian

Dalam mengembangkan model persediaan dan penetapan harga *single*-produk menjadi multi-produk tentu terdapat beberapa hal yang perlu dipertimbangkan. Salah satunya adalah fungsi permintaan. Apa saja yang mempengaruhi permintaan suatu produk jika terdapat lebih dari satu produk. Apakah harga produk lain mempengaruhi permintaan konsumen. Shavandi et al., (2012) mengkategorikan produk ke dalam beberapa jenis, diantaranya adalah independen dan substitusi sehingga formula dalam penelitian ini akan mengikuti kategori produk tersebut.

a. Fungsi permintaan

Pada penelitian ini, fungsi permintaan dari setiap produk mempertimbangkan harga jual, *displayed stock*, *expiration date*, serta efek harga dari produk lain untuk kategori produk substitusi. Fungsi ini terbentuk berdasarkan kombinasi model permintaan dari Feng et al., (2017) dan Shavandi et al., (2012). Fungsi permintaan tersebut diformulasikan seperti di bawah ini:

- Produk independen, yaitu produk dengan permintaan yang tidak mempertimbangkan produk lain dapat diformulasikan sebagai berikut

$$D(t, P) = \alpha w^\beta \frac{m-t}{m} e^{-\lambda P} ; 0 \leq t \leq t_1 \leq T \quad (4.10)$$

- Produk substitusi, yaitu produk i yang memiliki produk pengganti j dapat diformulasikan sebagai berikut:

$$D(t, P) = \alpha w^\beta \frac{m-t}{m} e^{-\lambda P_i + n_j P_j}; 0 \leq t \leq t_1 \leq T; i, j = 1, 2, \dots, n \quad (4.11)$$

- b. Fungsi tujuan

$$\text{Max } Z = \sum_{i=1}^n \text{Profit} \quad (4.12)$$

Model ini tidak mempertimbangkan *planning horizon* karena waktu siklus untuk setiap produk berbeda dan siklus akan berulang. Pada model ini, *output profit* yang dihasilkan adalah per tahun dengan asumsi periode waktu 1 tahun adalah 365 hari. Formula *profit* dibedakan berdasarkan kategori produk, yaitu:

- *Profit* produk independen

$$\text{Profit}(Q, P, T) = \frac{1}{T} \left\{ P(Q - E) + sE - cQ - o - h \left[\alpha w^\beta e^{-\lambda P} \left(\frac{t_1^3}{6m} - \frac{t_1^2}{2} \right) + Qt_1 \right] - \frac{h}{2}(w + E)(T - t_1) \right\} \quad (4.13)$$

Dengan Batasan:

- a. *Order quantity* harus lebih besar atau sama dengan ukuran rak dalam unit

$$Q = w + \frac{\alpha w^\beta e^{-\lambda_i P_i}}{2m} \left[m^2 - (m - T)^2 - \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)e^{-\lambda_i P_i}} \right] \geq w \quad (4.14)$$

- b. Waktu ketika *inventory level* mencapai kuantitas ukuran rak/tidak ada stok di gudang (t_1) harus lebih besar atau sama dengan 0.

$$t_1 = m - \sqrt{(m - T)^2 + \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)}} \geq 0 \quad (4.15)$$

- *Profit* produk substitusi

$$\text{Profit}(Q, P, T) = \frac{1}{T} \sum_{i=1}^n \left\{ P_i(Q_i - E_i) + s_i E_i - c_i Q_i - o_i - h_i \left[\alpha_i w_i^{\beta_i} e^{-\lambda_i P_i + n_j P_j} \left(\frac{t_{i1}^3}{6m_i} - \frac{t_{i1}^2}{2} \right) + Q_i t_{i1} \right] - \frac{h_i}{2}(w_i + E_i)(T_i - t_{i1}) \right\} \quad (4.16)$$

Dengan batasan:

- a. *Order quantity* harus lebih besar atau sama dengan ukuran rak dalam unit.

$$Q = w + \frac{\alpha w^\beta e^{-\lambda_i P_i + n_j P_j}}{2m} \left[m^2 - (m - T)^2 - \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)e^{-\lambda_i P_i + n_j P_j}} \right] \quad (4.17)$$

- b. Waktu ketika *inventory level* mencapai kuantitas ukuran rak/tidak ada stok di gudang (t_1) harus lebih besar atau sama dengan 0.

$$t_1 = m - \sqrt{(m - T)^2 + \frac{2m(w^{1-\beta} - E^{1-\beta})}{\alpha(1-\beta)}} \quad (4.18)$$

Fungsi tujuan di atas secara keseluruhan dibatasi oleh beberapa batasan, yaitu:

$$Q \geq w \quad (4.19)$$

$$t_1 \geq 0 \quad (4.20)$$

$$\mu_i - a_i \sigma_i' \leq P_i \leq \mu_i + a_i \sigma_i' \quad (4.21)$$

$$0 \leq E \leq w, \text{ dan } 0 \leq t_1 \leq T \leq m \quad (4.22)$$

- c. Notasi

Z : fungsi tujuan / fungsi objektif

T : *replishment cycle time* dalam unit waktu

Q : *order quantity* dalam unit

E : persediaan akhir dalam unit ; $E \geq 0$

P: harga per unit

s : harga sisa (*salvage*) per unit

o : biaya pesan

h : biaya penyimpanan

c : harga beli

t_1 : waktu ketika *inventory level* mencapai w

α : jumlah maksimum dari *potential customer*

β : sensitivitas ruang rak (*shelf space*) terhadap *demand*

λ : sensitivitas harga terhadap *demand*

w: ukuran ruang rak dalam unit

m: usia maksimal produk

t: periode waktu

μ_i : harga rata-rata produk i

σ_i' : standar deviasi harga produk i

a_i : koefisien dengan $a_i \geq 1$

n: sensitivitas harga produk substitusi terhadap *demand* (λ dari produk j)

4.3 Skenario

Dalam penelitian ini, penulis melakukan dua skenario untuk menemukan nilai *profit* optimal. Kedua skenario tersebut dilakukan untuk produk independen dan produk substitusi. Skenario yang dilakukan adalah mempertimbangkan kapasitas gudang yang ditambahkan pada batasan model. Hal ini didasari penelitian yang di yang dilakukan oleh Zoller (1977). Skenario tersebut dijelaskan sebagai berikut:

5.3.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

Pada skenario ini, penulis mempertimbangkan kapasitas gudang sebagai batasan untuk masing-masing produk sehingga batasan pada skenario ini adalah:

$$Q_i \leq \hat{c} \quad (4.23)$$

Dimana:

Q_i : order quantity produk i dalam unit

\hat{c} : kapasitas gudang yang tersedia

Kemudian, model skenario 1 untuk masing-masing kategori produk, yaitu:

a. Produk Independen

Fungsi permintaan yang digunakan pada kategori produk independen tidak mempertimbangkan keberadaan produk lain sehingga *profit* multi-produk dihasilkan berdasarkan produk masing-masing. Berikut formula *profit* seperti pada persamaan 4.13:

$$P(E, P, T) = \frac{1}{T} \left\{ P(Q - E) + sE - cQ - o - h \left[\alpha w^\beta e^{-\lambda P} \left(\frac{t_1^3}{6m} - \frac{t_1^2}{2} \right) + Q t_1 \right] - \frac{h}{2} (w + E)(T - t_1) \right\} \quad (4.24)$$

Formula di atas selanjutnya menjadi *input* pada proses maksimasi *total profit* multi-produk ke dalam fungsi objektif pada formula 4.12 sebagai berikut:

$$\text{Max } Z = \sum_{i=1}^n \text{Profit} \quad (4.25)$$

Sehingga, batasan pada kategori produk ini, yaitu:

$$Q \geq w \quad (4.26)$$

$$t_1 \geq 0 \quad (4.27)$$

$$\mu_i - a_i \sigma_i' \leq P_i \leq \mu_i + a_i \sigma_i' \quad (4.28)$$

$$Q_i \leq \hat{c}; 0 \leq E \leq w, \text{ dan } 0 \leq t_l \leq T \leq m \quad (4.29)$$

b. Produk Substitusi

Berbeda dengan kategori produk sebelumnya, fungsi permintaan yang digunakan pada kategori produk substitusi mempertimbangkan keberadaan produk lain yang merupakan substitusi dari produk tersebut sehingga *profit* multi-produk dihasilkan berdasarkan efek dari kedua produk tersebut. Sehingga, formula *profit* seperti pada persamaan 4.16:

$$\text{Profit}(E, P, T) = \frac{1}{T} \sum_{i=1}^n \left\{ P_i(Q_i - E_i) + s_i E_i - c_i Q_i - o_i - h_i \left[\alpha_i w_i^{\beta_i} e^{-\lambda_i P_i + n_j P_j} \left(\frac{t_{i1}^3}{6m_i} - \frac{t_{i1}^2}{2} \right) + Q_i t_{i1} \right] - \frac{h_i}{2} (w_i + E_i)(T_i - t_{i1}) \right\} \quad (4.30)$$

Formula di atas selanjutnya menjadi *input* pada proses maksimasi *total profit* multi-produk ke dalam fungsi objektif pada formula 4.12 sebagai berikut:

$$\text{Max } Z = \sum_{i=1}^n \text{Profit} \quad (4.31)$$

Sehingga, batasan pada kategori produk ini, yaitu:

$$Q \geq w \quad (4.32)$$

$$t_1 \geq 0 \quad (4.33)$$

$$\mu_i - a_i \sigma_i' \leq P_i \leq \mu_i + a_i \sigma_i' \quad (4.34)$$

$$Q_i \leq \hat{c}; 0 \leq E \leq w, \text{ dan } 0 \leq t_l \leq T \leq m \quad (4.35)$$

5.3.1 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (aggregate)

Pada skenario ini, penulis mempertimbangkan kapasitas gudang sebagai batasan untuk keseluruhan (*aggregate*) produk sehingga batasan pada skenario ini adalah:

$$\sum_{i=1}^n Qi \leq \hat{c} \quad (4.36)$$

Dimana:

Q_i : order quantity produk i dalam unit

\hat{c} : kapasitas gudang yang tersedia

Kemudian, model skenario 2 untuk masing-masing kategori produk, yaitu:

a. Produk Independen

Fungsi permintaan yang digunakan pada kategori produk independen tidak mempertimbangkan keberadaan produk lain sehingga *profit* multi-produk dihasilkan berdasarkan produk masing-masing. Berikut formula *profit* seperti pada persamaan 4.13:

$$P(E, P, T) = \frac{1}{T} \left\{ P(Q - E) + sE - cQ - o - h \left[\alpha w^\beta e^{-\lambda P} \left(\frac{t_1^3}{6m} - \frac{t_1^2}{2} \right) + Qt_1 \right] - \frac{h}{2} (w + E)(T - t_1) \right\} \quad (4.37)$$

Formula di atas selanjutnya menjadi *input* pada proses maksimasi *total profit* multi-produk ke dalam fungsi objektif pada formula 4.12 sebagai berikut:

$$\text{Max } Z = \sum_{i=1}^n \text{Profit} \quad (4.38)$$

Sehingga, batasan pada kategori produk ini, yaitu:

$$Q \geq w \quad (4.39)$$

$$t_1 \geq 0 \quad (4.40)$$

$$\sum_{i=1}^n Qi \leq \hat{c} \quad (4.41)$$

$$\mu_i - a_i \sigma_i' \leq P_i \leq \mu_i + a_i \sigma_i' \quad (4.42)$$

$$0 \leq E \leq w, \text{ dan } 0 \leq t_l \leq T \leq m \quad (4.43)$$

b. Produk Substitusi

Berbeda dengan kategori produk sebelumnya, fungsi permintaan yang digunakan pada kategori produk substitusi mempertimbangkan keberadaan produk lain yang merupakan substitusi dari produk tersebut sehingga *profit* multi-produk dihasilkan berdasarkan efek dari kedua produk tersebut. Sehingga, formula *profit* seperti pada persamaan 4.16:

$$Profit(E, P, T) = \frac{1}{T} \sum_{i=1}^n \left\{ P_i(Q_i - E_i) + s_i E_i - c_i Q_i - o_i - h_i \left[\alpha_i w_i \beta_i e^{-\lambda_i P_i + n_j P_j} \left(\frac{t_{i1}^3}{6m_i} - \frac{t_{i1}^2}{2} \right) + Q_i t_{i1} \right] - \frac{h_i}{2} (w_i + E_i)(T_i - t_{i1}) \right\} \quad (4.44)$$

Formula di atas selanjutnya menjadi *input* pada proses maksimasi *total profit* multi-produk ke dalam fungsi objektif pada formula 4.12 sebagai berikut:

$$Max Z = \sum_{i=1}^n Profit \quad (4.45)$$

Sehingga, batasan pada kategori produk ini, yaitu:

$$Q \geq w \quad (4.46)$$

$$t_1 \geq 0 \quad (4.47)$$

$$\mu_i - \alpha_i \sigma_i' \leq P_i \leq \mu_i + \alpha_i \sigma_i' \quad (4.48)$$

$$\sum_{i=1}^n Q_i \leq \hat{c} \quad (4.49)$$

$$0 \leq E \leq w, \text{ dan } 0 \leq t_l \leq T \leq m \quad (4.50)$$

4.4 Verifikasi

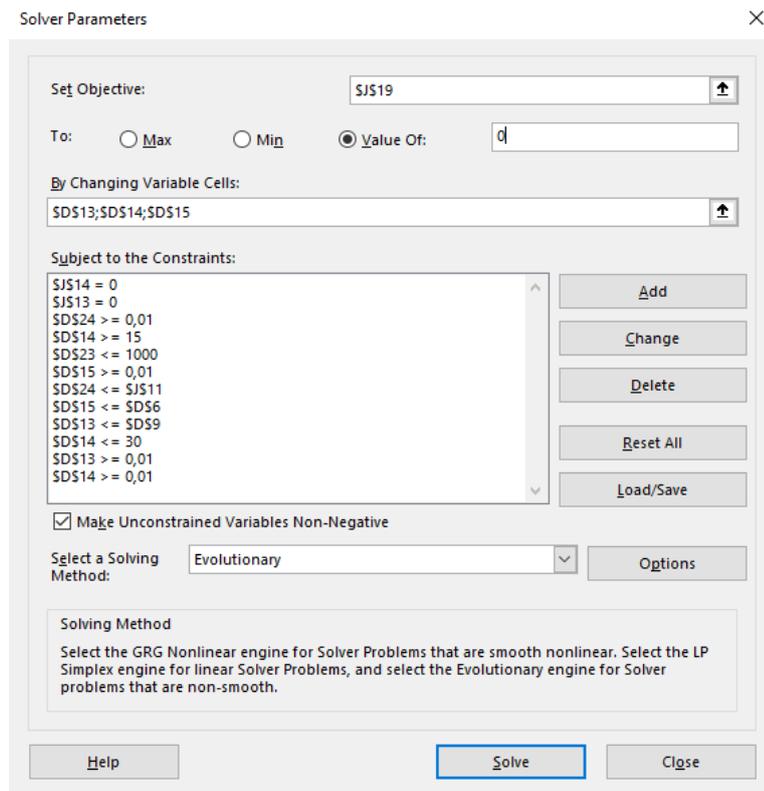
Sebelum melakukan optimasi menggunakan metode yang telah dipilih, model yang telah dikembangkan harus diverifikasi terlebih dahulu. Verifikasi ini dilakukan untuk mengetahui apakah model matematika yang telah dikembangkan telah sesuai dengan konseptual model atau logika diagram alur. Verifikasi dilakukan menggunakan 2 cara, yaitu: 1. verifikasi menggunakan *solver* pada excel; dan 2. verifikasi menggunakan metode metaheuristik Algoritma *Artificial Bee Colony* dengan bahasa pemrograman “*Phyton*” dan *software* “*Anaconda Navigator 3*”.

Verifikasi menggunakan cara 1 dilakukan dengan menggunakan *data input* berdasarkan data pada percobaan numerik jurnal Feng et al., (2017) seperti pada

tabel 4.1. Sedangkan untuk parameter *solver* yang digunakan dalam verifikasi dapat dilihat gambar 4.3 Jumlah produk yang digunakan pada verifikasi ini adalah 1 produk karena *solver* pada excel tidak mampu menjalankan model dengan jumlah batasan yang banyak. Kemudian, setelah *data input* dan parameter selesai di-*input*, *solver* siap dijalankan. Hasil dari verifikasi tersebut dapat dilihat pada tabel 4.2. Setelah itu, hasil tersebut dibandingkan dengan hasil percobaan numerik pada penelitian yang dilakukan oleh Feng et al., (2017) seperti pada tabel 4.2. Hasilnya, *output* yang dihasilkan antara verifikasi dan Feng et al., (2017) tidak memiliki perbedaan yang signifikan sehingga model telah terverifikasi.

Tabel 4. 1 Data Input Model Untuk Verifikasi

Data Input Model				
c	<i>purchasing cost</i>	=	15,5	Rp1.000/ unit
h	<i>holding cost</i>	=	5	Rp1.000/ unit/ year
m	<i>maximum life time</i>	=	0,04	year
o	<i>order cost</i>	=	8	Rp1.000/order
s	<i>salvage price</i>	=	12	Rp1.000/unit
w	<i>size of shelf space</i>	=	50	unit
μ	Harga rata-rata	=	27,5	Rp1.000/unit
a	Koefisien	=	1	
σ	Standar deviasi produk i	=	7,5	1.000
α	<i>potential customers</i>	=	6000	
β	<i>shelf space efficiency of demand</i>	=	0,6	
λ	<i>price efficiency of demand</i>	=	0,1	
\hat{c}	Kapasitas gudang	=	1000	



Gambar 4. 3 Parameter Solver

Tabel 4. 2 Perbandingan *Output* Model Menggunakan *Solver* vs *Output* Jurnal Feng et al., (2017)

Perbandingan Output Model		
	Model Penelitian ini	Model Feng et al. (2017)
E1	6 unit	6,5156 unit
P1	\$27	\$28,19
Q1	43 units	39,6586 unit
T1	8,76 hari/0,024 years	0,023 years
<i>Profit</i> 1	\$8.230	\$8.366,46

Verifikasi menggunakan cara 2 dilakukan menggunakan skenario 1 produk independen dengan kapasitas gudang dibuat besar agar hasil dapat maksimal. Anaconda seperti software lainnya akan memberikan peringatan *error* jika ketidaksesuaian algoritma terjadi. Dalam verifikasi ini sudah tidak ada notifikasi error yang muncul sehingga dapat dikatakan bahwa algoritma sudah benar. *Data input* parameter yang digunakan pada verifikasi ini juga menggunakan data *numerical example* dari penelitian Feng et al., (2017) sehingga produk yang

diverifikasi hanya 1 produk. Data tersebut di-input menggunakan Bahasa pemrograman “Python” dengan software “Anaconda Navigator”. Data input yang digunakan tersebut dapat dilihat pada gambar 4.4. pada gambar tersebut, data yang berasal dari penelitian Feng et al., (2017) adalah sebelah kiri (produk 1). Sedangkan untuk data sebelah kanan adalah asumsi. Setelah itu, model di-running dengan menggunakan 5000 iterasi. Hasil dari running model tersebut dapat dilihat pada gambar 4.5.

```

# Parameter
c = [20, 13] # Unit cost per unit
h = [5, 4] # Holding cost per unit
m = [15, 11] # Maximum lifetime of product
o = [10, 7] # Order cost per order
s = [10, 10] # Salvage price
w = [25, 50] # Shelf capacity
pm = [22.5, 25] # Price mean
ps = [7.5, 10] # Price std
alpha = [6000, 3000] # Maximum potential customer
beta = [0.6, 0.2] # Display stock weight
gamma = [0.1, 0.2] # Price weight

# Scenario
product = 'independent' #Product type (independent, substitute)
warehouse = 'single' #warehouse type (single, mixed)
cap = [1000, 150] #Warehouse capacity

# Hyper Parameter
n = 5000 #Iteration
P = 30 #Population size
P_ratio = 0.5 #Employed, Onlooker ratio
E_bee = int(P * P_ratio) #Employed bee
O_bee = int(P * P_ratio) #Onlooker bee
limit = 5 #Limit
counter = np.array([0]*E_bee) #Limit counter
D = 6 #Dimension
space = np.array([[0, cap[0]], # Q1 (θ-capacity)
                  [0, cap[1]], # Q2 (θ-capacity)
                  [pm[0] - ps[0], pm[0] + ps[0]], # P1 [mean-std, mean+std]
                  [pm[1] - ps[1], pm[1] + ps[1]], # P2 [mean-std, mean+std]
                  [0, m[0]], # T1 (θ - max lifetime)
                  [0, m[1]]]) # T2 (θ - max lifetime)
#Initial searching space

```

Gambar 4. 4 Data Input / Parameter Model

```

{'E1': 6,
 'E2': 46,
 'P1': 28,
 'P2': 15,
 'Q1': 40,
 'Q2': 50,
 'T1': 8,
 'T2': 11,
 'Profit1': 9130.0,
 'Profit2': -4778.636363636364}

```

Gambar 4. 5 Output yang Dihasilkan

Setelah itu, hasil tersebut dibandingkan dengan hasil *numerical example* dari penelitian Feng et al., (2017). Pada penelitian Feng et al., (2017), *output* yang dihasilkan, yaitu: $E = 6,5156$ unit, $P = \$28,19$, $Q = 39,6586$, $T = 0,023$ tahun atau 8 hari, dan *profit* = \$8.366,46. Berdasarkan hasil dari perbandingan tersebut, *output* yang dihasilkan tidak terlalu berbeda bahkan terdapat *output* yang sama persis seperti pada tabel 4.3. Oleh karena itu, model yang telah dikembangkan ini telah terverifikasi.

Tabel 4. 3 Perbandingan Output Model

Perbandingan Output Model		
	Model Penelitian ini	Modeng Feng et al. (2017)
E1	6 unit	6,5156 unit
P1	\$28	\$28,19
Q1	40 units	39,6586 unit
T1	8 hari/0,0219 years	0,023 years
<i>Profit1</i>	\$9.130	\$8.366,46

BAB 5

ANALISIS DAN PEMBAHASAN

Bab ini berisi mengenai percobaan numerik dari model yang telah dirancang sebelumnya dengan menggunakan metode metaheuristik *Artificial Bee Colony* (ABC). Bab ini juga menampilkan beberapa skenario perubahan parameter untuk melihat efek dari perubahan parameter tersebut atau disebut analisis sensitivitas. Tujuan dilakukan analisis sensitivitas tersebut adalah untuk melihat pengaruh perubahan parameter terhadap fungsi tujuan yang diharapkan. Selain itu, analisis hasil dapat dijadikan penarikan kesimpulan pada bab selanjutnya.

5.1. Parameter dan *Constraint*, *Hyper Parameter*, *Fitness*, dan Algoritma *Artificial Bee Colony*

Percobaan numerik dilakukan dengan menggunakan Algoritma *Artificial Bee Colony* dengan Bahasa pemrograman “*Python*” dan *software* “*Anaconda Navigator 3*” serta “PC Lenovo ideapad 320 intel core i3”.

5.1.1. Parameter

Parameter yang digunakan pada percobaan numerik penelitian ini dibedakan berdasarkan skenario dan tipe produk seperti pada Bab IV. Parameter yang digunakan didasari pada percobaan numerik penelitian Feng et al., (2017), *desk research*, serta penyesuaian penelitian sebelumnya hal ini dikarenakan penelitian ini merupakan pengembangan model dengan tujuan optimalisasi. Parameter yang digunakan pada percobaan numerik ini, yaitu:

- Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

Percobaan numerik pada skenario 1, produk memiliki kapasitas gudang sendiri-sendiri (*single*) dan tidak mempertimbangkan kapasitas gudang produk lain yang dibagi ke dalam 2 tipe produk, yaitu: produk independen dan produk substitusi. Parameter yang digunakan pada skenario ini dapat dilihat pada tabel 5.1 untuk produk independen (produk 1 adalah alpukat lokal dan produk 2 adalah

apel lokal) dan tabel 5.2 untuk produk substitusi (produk 1 adalah apel impor dan produk 2 adalah apel lokal).

Tabel 5. 1 Data Input Produk Independen Skenario 1

Produk 1 (Alpukat Lokal)				Produk 2 (Apel Lokal)			
c	purchasing cost	=	15.500 Rp/ unit	c	purchasing cost	=	13.000 Rp/ unit
h	holding cost	=	5.000 Rp/ unit/ year	h	holding cost	=	4.000 Rp/ unit/ year
m	maximum life time	=	0,04 year	m	maximum life time	=	0,03 year
o	order cost	=	8.000 Rp/order	o	order cost	=	7.000 Rp/order
s	salvage price	=	12.000 Rp/unit	s	salvage price	=	10 Rp/unit
w	size of shelf space	=	50 unit	w	size of shelf space	=	50 unit
μ	Harga rata-rata	=	27.500 Rp/unit	μ	Harga rata-rata	=	25.000 Rp/unit
a	Koefisien	=	1	a	Koefisien	=	1
σ	Standar deviasi produk i	=	7.500	σ	Standar deviasi produk i	=	10
α	potential customers	=	10000	α	potential customers	=	10000
β	shelf space efficiency of demand	=	0,6	β	shelf space efficiency of demand	=	0,5
λ	price efficiency of demand	=	0,1	λ	price efficiency of demand	=	0,2
\hat{c}	Kapasitas gudang	=	150	\hat{c}	Kapasitas gudang	=	150

Tabel 5. 2 Data Input Produk Substitusi Skenario 1

Produk 1 (Apel Impor)				Produk 2 (Apel Lokal)			
c	purchasing cost	=	33.000 Rp/ unit	c	purchasing cost	=	13.000 Rp/ unit
h	holding cost	=	7.000 Rp/ unit/ year	h	holding cost	=	4.000 Rp/ unit/ year
m	maximum life time	=	0,04 year	m	maximum life time	=	0,03 year
o	order cost	=	17.000 Rp/order	o	order cost	=	7.000 Rp/order
s	salvage price	=	18.000 Rp/unit	s	salvage price	=	10 Rp/unit
w	size of shelf space	=	50 unit	w	size of shelf space	=	50 unit
μ	Harga rata-rata	=	45.000 Rp/unit	μ	Harga rata-rata	=	25.000 Rp/unit
a	Koefisien	=	1	a	Koefisien	=	1
σ	Standar deviasi produk i	=	10	σ	Standar deviasi produk i	=	10
α	potential customers	=	3.000	α	potential customers	=	10.000
β	shelf space efficiency of demand	=	0,6	β	shelf space efficiency of demand	=	0,5
λ	price efficiency of demand	=	0,1	λ	price efficiency of demand	=	0,2

\hat{c}	kapasitas gudang	=	150		\hat{c}	kapasitas gudang	=	150
$e = 2,71828182845905$								

- Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

Percobaan numerik pada skenario 2, kapasitas gudang yang digunakan adalah kumulatif (*aggregate*) sehingga mempertimbangkan produk lain yang dibagi ke dalam 2 tipe produk, yaitu: produk independen dan produk substitusi. Parameter yang digunakan pada skenario ini dapat dilihat pada tabel 5.3 (produk 1 adalah alpukat lokal dan produk 2 adalah apel lokal) untuk produk independen dan tabel 5.4 untuk produk substitusi (produk 1 adalah apel impor dan produk 2 adalah apel lokal).

Tabel 5. 3 Data Input Produk Independen Skenario 2

Produk 1 (Alpukat Lokal)				Produk 2 (Apel Lokal)			
c	purchasing cost	=	15.500 Rp/ unit	c	purchasing cost	=	13.000 Rp/ unit
h	holding cost	=	5.000 Rp/ unit/ year	h	holding cost	=	4.000 Rp/ unit/ year
m	maximum life time	=	0,04 year	m	maximum life time	=	0,03 year
o	order cost	=	8.000 Rp/order	o	order cost	=	7.000 Rp/order
s	salvage price	=	12.000 Rp/unit	s	salvage price	=	10 Rp/unit
w	size of shelf space	=	50 unit	w	size of shelf space	=	50 unit
μ	Harga rata-rata	=	27.500 Rp/unit	μ	Harga rata-rata	=	25.000 Rp/unit
a	Koefisien	=	1	a	Koefisien	=	1
σ	Standar deviasi produk i	=	7.500	σ	Standar deviasi produk i	=	10
α	potential customers	=	10.000	α	potential customers	=	10.000
β	shelf space efficiency of demand	=	0,6	β	shelf space efficiency of demand	=	0,5
λ	price efficiency of demand	=	0,1	λ	price efficiency of demand	=	0,2
$\hat{c} =$ kapasitas gudang keseluruhan (<i>aggregate</i>) = 300							

Parameter tersebut akan digunakan sebagai *data input* pada percobaan numerik penelitian ini. Secara algoritma, parameter yang digunakan dapat dilihat pada gambar 5.1 dengan contoh parameter dari salah satu skenario dan tipe produk di atas. Sedangkan algoritma *constraint* yang digunakan dalam

memecahkan permasalahan dengan *Artificial Bee Colony* dapat dilihat pada gambar 5.2.

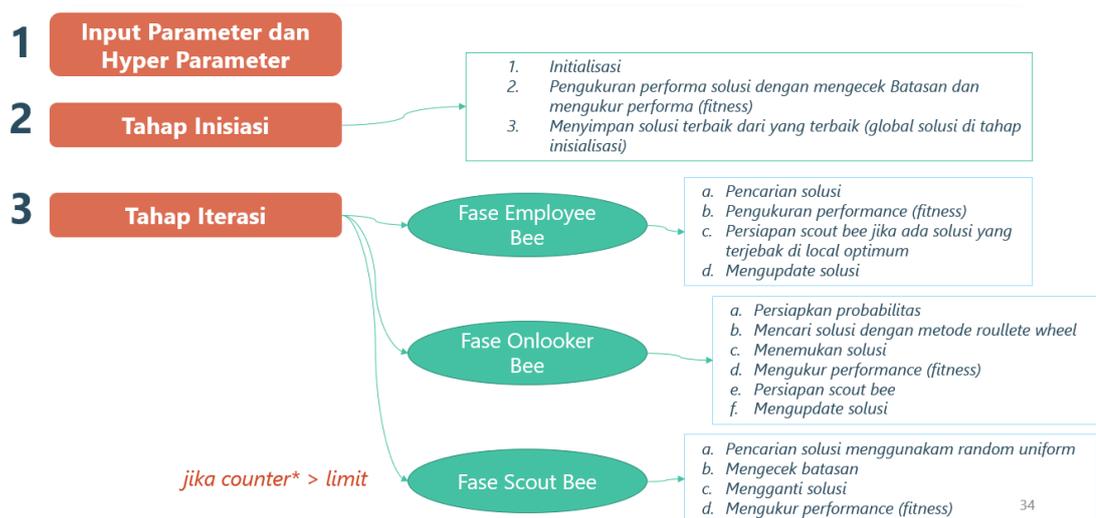
Tabel 5. 4 Data Input Produk Substitusi Skenario 2

Produk 1 (Apel Impor)					Produk 2 (Apel Lokal)				
c	purchasing cost	=	33.000	Rp/ unit	c	purchasing cost	=	13.000	Rp/ unit
h	holding cost	=	7.000	Rp/ unit/ year	h	holding cost	=	4.000	Rp/ unit/ year
m	maximum life time	=	0,04	year	m	maximum life time	=	0,03	year
o	order cost	=	17.000	Rp/order	o	order cost	=	7.000	Rp/order
s	salvage price	=	18.000	Rp/unit	s	salvage price	=	10	Rp/unit
w	size of shelf space	=	50	unit	w	size of shelf space	=	50	unit
μ	Harga rata-rata	=	45.000	Rp/unit	μ	Harga rata-rata	=	25.000	Rp/unit
a	Koefisien	=	1		a	Koefisien	=	1	
σ	Standar deviasi produk i	=	10		σ	Standar deviasi produk i	=	10	
α	potential customers	=	3.000		α	potential customers	=	10.000	
β	shelf space efficiency of demand	=	0,6		β	shelf space efficiency of demand	=	0,5	
λ	price efficiency of demand	=	0,1		λ	price efficiency of demand	=	0,2	
\hat{c}	kapasitas gudang	=	150		\hat{c}	kapasitas gudang	=	150	
\hat{c} = kapasitas gudang keseluruhan (aggregate) = 300									
$e = 2,71828182845905$									

5.1.2. Tahapan Percobaan Numerik dengan Algoritma Artificial Bee Colony (ABC)

Dalam mencari solusi optimal pada penelitian ini, tahapan percobaan numerik dengan menggunakan Algoritma *Artificial Bee Colony* (*coding* terlampir) terdiri dari 3 tahapan yang dapat dilihat pada gambar 5.1. Tahapan tersebut, yaitu:

1. Tahap input parameter dan *hyper parameter*
2. Tahap inisiasi, yaitu: inialisasi, pengukuran performa, dan *update* solusi.
3. Tahap iterasi terdiri dari 3 fase, yaitu: fase *employed bee*, fase *onlooker bee*, dan fase *scout bee*.



*solusi yang tidak lebih baik daripada solusi sebelumnya

Gambar 5. 1 Tahapan Percobaan Numerik dengan *Algoritma Artificial Bee Colony*

5.2. Percobaan Numerik

Percobaan numerik pada penelitian ini dilakukan dengan menggunakan metode metaheuristik. Percobaan numerik ini dilakukan dengan 2 skenario yang dibagi berdasarkan kategori produk, yaitu: produk independen dan produk substitusi. Skenario ini dilakukan dengan mempertimbangkan kapasitas gudang sebagai batasan seperti yang telah dibahas pada Bab IV. Percobaan numerik pada skenario 1, produk memiliki kapasitas gudang sendiri-sendiri (*single*) dan tidak mempertimbangkan kapasitas gudang produk lain. Sedangkan percobaan numerik pada skenario 2, kapasitas gudang yang digunakan adalah kumulatif (*aggregate*) sehingga mempertimbangkan produk lain.

Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

Berikut hasil percobaan numerik yang dilakukan dengan batasan kapasitas gudang *single*.

a. Produk Independen

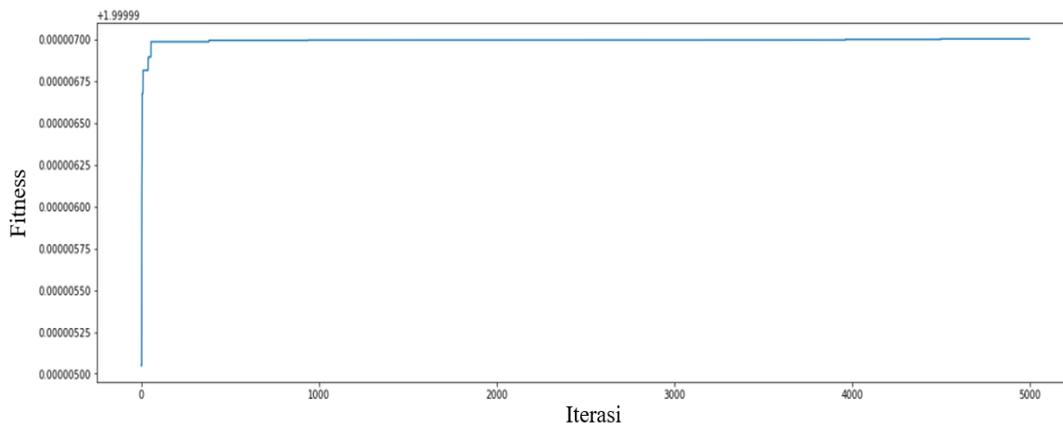
Hasil percobaan numerik menggunakan algoritma *Artificial Bee Colony* skenario 1 pada produk independen dapat dilihat pada tabel 5.5. Pada tabel tersebut, dapat dilihat bahwa *total profit* produk 1 dan produk 2 adalah

Rp59.067.000,00 dengan *profit* produk 1 adalah Rp57.783.120,00 dan *profit* produk 2 adalah Rp1.283.880,00.

Tabel 5. 5 Hasil Percobaan Numerik Skenario 1 Produk Independen

	P (Rp1.000)		Q (unit)		T (hari)		Profit (Rp1.000)		Total Profit (Rp1.000)
	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
Output	25	16	91	50	4	9	57.783,1	1.283,88	59.067

Sedangkan grafik iterasi *fitness* yang dilakukan algoritma *Artificial Bee Colony* dengan *software Python* dapat dilihat pada gambar 5.2. Grafik ini menunjukkan bahwa *fitness* per iterasi yang dilakukan mengalami peningkatan yang menunjukkan bahwa model yang diteliti mencari solusi dengan *profit* yang lebih tinggi sehingga fungsi objektif dari model ini adalah maksimasi.



Gambar 5. 2 Grafik Iterasi *Fitness* Skenario 1 Produk Independen

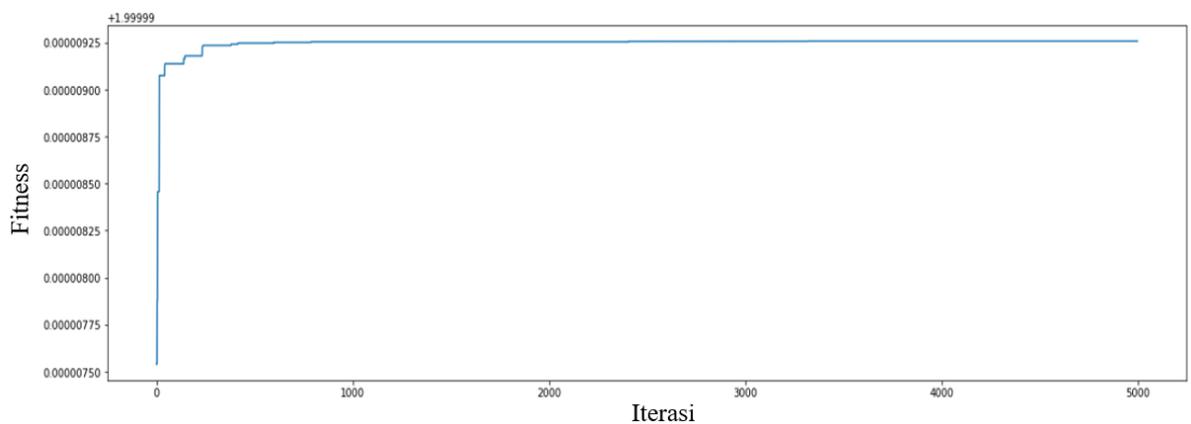
b. Produk Substitusi

Hasil percobaan numerik menggunakan algoritma *Artificial Bee Colony* skenario 1 pada produk substitusi dapat dilihat pada tabel 5.6. Pada tabel tersebut, dapat dilihat bahwa *total profit* produk 1 dan produk 2 adalah Rp1.341.620.670,00 dengan *profit* produk 1 adalah Rp1.198.295.000,00 dan *profit* produk 2 adalah Rp143.325.670,00.

Tabel 5. 6 Hasil Percobaan Numerik Skenario 1 Produk Substitusi

	P (Rp1.000)		Q (unit)		T (hari)		Profit (Rp1.000)		Total Profit (Rp1.000)
	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
Output	55	31	150	94	1	3	1.198.295	143.326	1.341.620,7

Sedangkan grafik iterasi *fitness* yang dilakukan algoritma *Artificial Bee Colony* dengan *software Phyton* dapat dilihat pada gambar 5.3. Grafik ini menunjukkan bahwa *fitness* per iterasi yang dilakukan mengalami peningkatan yang menunjukkan bahwa model yang diteliti mencari solusi dengan *profit* yang lebih tinggi sehingga fungsi objektif dari model ini adalah maksimasi.



Gambar 5. 3 Grafik Iterasi *Fitness* Skenario 1 Produk Substitusi

Skenario 2: Batasan kapasitas gudang untuk seluruh produk (aggregate)

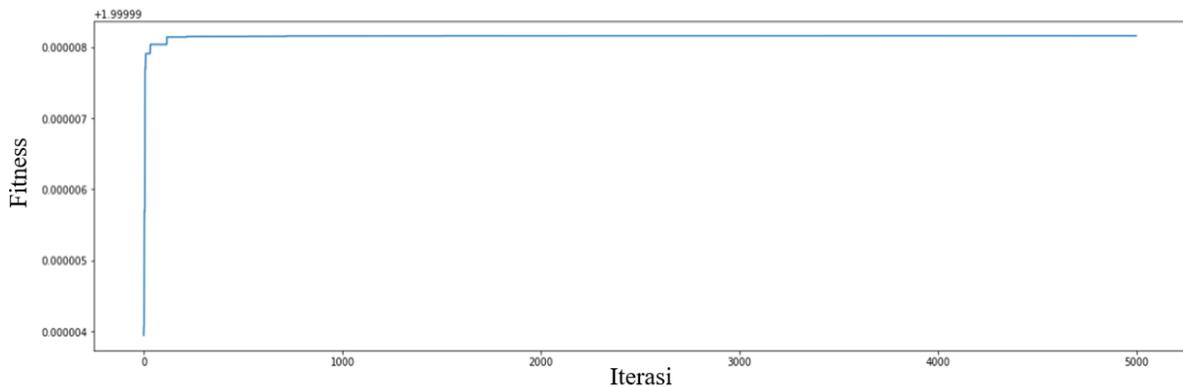
a. Produk Independen

Hasil percobaan numerik menggunakan algoritma *Artificial Bee Colony* skenario 2 pada produk independen dapat dilihat pada tabel 5.7. Pada tabel tersebut, dapat dilihat bahwa *total profit* produk 1 dan produk 2 adalah Rp59.069.280,00 dengan *profit* produk 1 adalah Rp57.995.000,00 dan *profit* produk 2 adalah Rp1.074.880,00.

Tabel 5. 7 Hasil Percobaan Numerik Skenario 2 Produk Independen

	P (Rp1.000)		Q (unit)		T (hari)		Profit (Rp1.000)		Total Profit (Rp1.000)
	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
<i>Output</i>	24	15	100	50	4	7	57.995	1.074,28	59.069,28

Sedangkan grafik iterasi *fitness* yang dilakukan algoritma *Artificial Bee Colony* dengan *software Python* dapat dilihat pada gambar 5.4. Grafik ini menunjukkan bahwa *fitness* per iterasi yang dilakukan mengalami peningkatan yang menunjukkan bahwa model yang diteliti mencari solusi dengan *profit* yang lebih tinggi sehingga fungsi objektif dari model ini adalah maksimasi.



Gambar 5. 4 Grafik Iterasi *Fitness* Skenario 2 Produk Independen

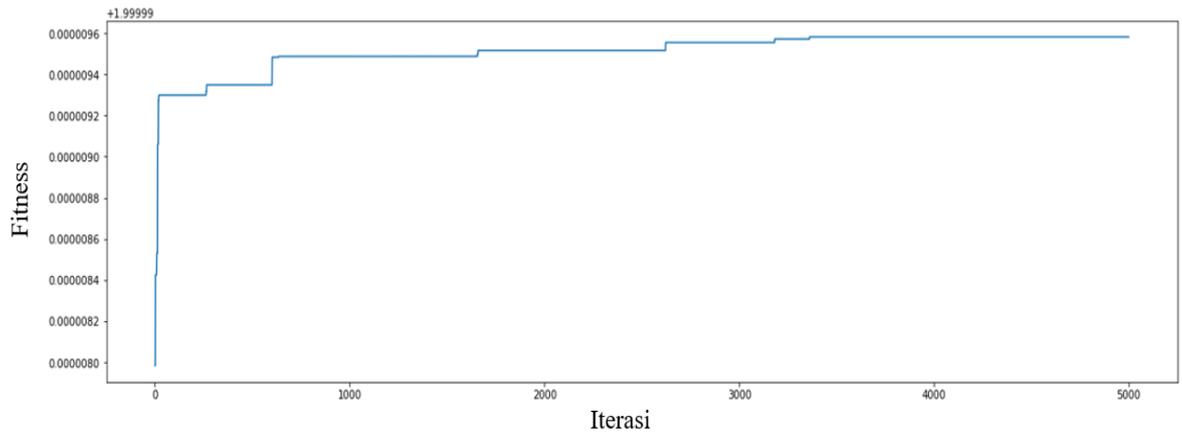
b. Produk Substitusi

Hasil percobaan numerik menggunakan algoritma *Artificial Bee Colony* skenario 2 pada produk substitusi dapat dilihat pada tabel 5.8. Pada tabel tersebut, dapat dilihat bahwa *total profit* produk 1 dan produk 2 adalah Rp2.334.390.290,00 dengan *profit* produk 1 adalah Rp2.332.686.990,00 dan *profit* produk 2 adalah Rp1.703.300,00.

Tabel 5. 8 Hasil Percobaan Numerik Skenario 2 Produk Substitusi

	P (Rp1.000)		Q (unit)		T (hari)		Profit (Rp1.000)		Total Profit (Rp1.000)
	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
<i>Output</i>	55	34	298	1	1	4	2.332.687	1.703,3	2.334.390,3

Sedangkan grafik iterasi *fitness* yang dilakukan algoritma *Artificial Bee Colony* dengan *software Python* dapat dilihat pada gambar 5.5. Grafik ini menunjukkan bahwa *fitness* per iterasi yang dilakukan mengalami peningkatan yang menunjukkan bahwa model yang diteliti mencari solusi dengan *profit* yang lebih tinggi sehingga fungsi objektif dari model ini adalah maksimasi.



Gambar 5. 5 Grafik Iterasi *Fitness* Skenario 2 Produk Substitusi

Berdasarkan hasil percobaan numerik di atas, dapat dirangkum perbandingan hasil yang didapatkan pada tabel 5.9. Berdasarkan tabel tersebut, dapat diketahui bahwa *total profit* dengan menggunakan skenario 2 (batasan kapasitas gudang untuk seluruh produk/*aggregate*) memiliki nilai lebih besar dibandingkan *total profit* dengan menggunakan skenario 1 (batasan kapasitas gudang untuk masing-masing produk).

Tabel 5. 9 Perbandingan *Profit* Per Skenario Per Tipe Produk

Perbandingan <i>Profit</i> Per Skenario Per Tipe Produk						
No.	Tipe Produk	Jenis Skenario	<i>Profit</i> (Rp1000)		<i>Total Profit</i> (Rp1.000)	Waktu Komputasi
			Produk 1	Produk 2		
1	Independen	Skenario 1	57.783,12	1.283,88	59.067	01:17
2		Skenario 2	57.995	1.074,28	59.069,28	01:30

3	Substitusi	Skenario 1	1.198.295	143.325,67	1.341.620,67	01:48
4		Skenario 2	2.332.686,99	1.703,30	2.334.390,29	01:59

5.3. Analisis Sensitivitas

Analisis sensitivitas menjelaskan gambaran efek perubahan yang dilakukan pada parameter terhadap *total profit*. Terdapat beberapa parameter yang akan dianalisis, yaitu: parameter α (*potential customer*), β (*shelf space efficiency of demand*), λ (*price efficiency of demand*), c (*purchasing cost*), h (*holding cost*), o (*ordering cost*), s (*salvage price*), dan w (*size of shelft space*). Analisis pada parameter tersebut akan dibahas pada sub sub bab di bawah.

5.3.1 Analisis Sensitivitas Parameter α

Salah satu parameter yang digunakan dalam analisis sensitivitas adalah α (*potential customer*). Analisis sensitivitas untuk parameter α dibagi ke dalam 2 skenario dengan 2 tipe produk. Analisis sensitivitas dilakukan dengan mengubah beberapa nilai pada parameter α terhadap produk yang diujikan. Parameter α diubah untuk mengetahui pengaruh parameter tersebut terhadap *total profit* pada 2 skenario. Berikut akan dibahas analisis sensitivitas untuk parameter α .

5.3.1.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

a. Produk Independen

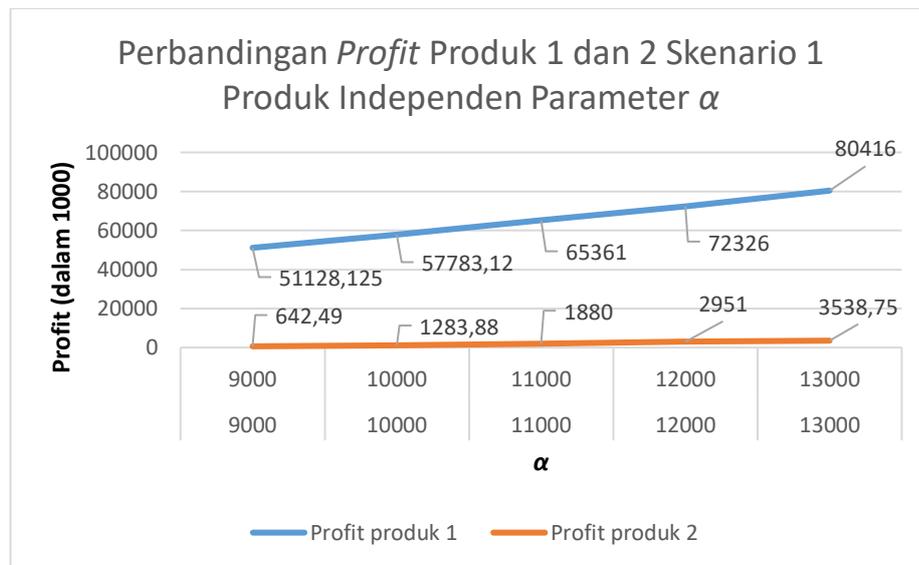
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 1.000 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.10.

Tabel 5. 10 *Output* Hasil Analisis Sensitivitas Parameter α

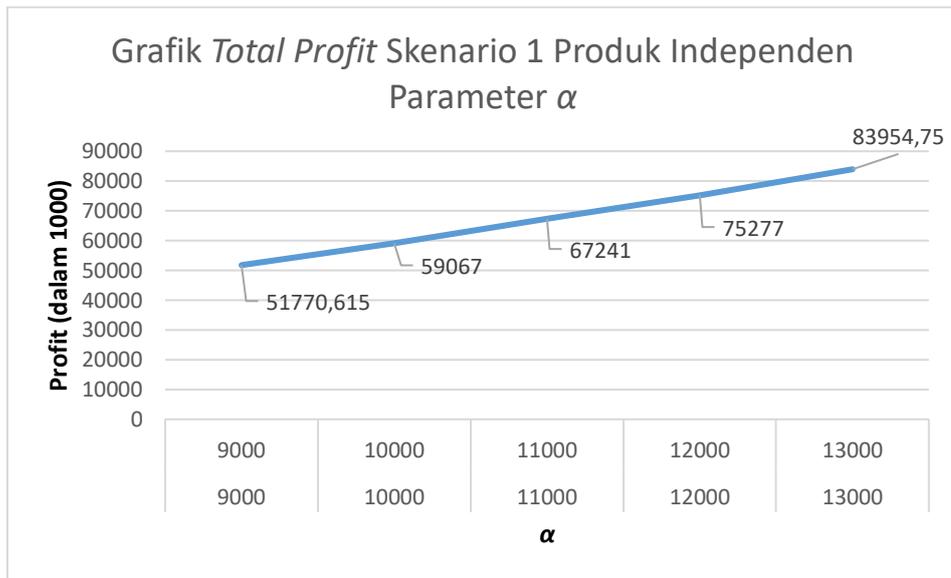
α										
Parameter		P (Rp1.000)		Q (unit)		T (hari)		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 1	produk 2	produk 1	produk 2	produk 1	produk 1	produk 2	produk 1	produk 2	
9.000	9.000	26	17	81	51	4	10	51.128,13	642,49	51.770,615

10.000	10.000	25	16	91	50	4	9	57.783,12	1.283,88	59.067
11.000	11.000	24	17	94	51	3	9	65.361	1.880	67.241
12.000	12.000	24	16	105	50	3	7	72.326	2.951	75.277
13.000	13.000	24	17	110	57	3	8	80.416	3.538,75	83.954,75

Garafik hasil analisis sensitivitas parameter α produk independen skenario 1 terlihat pada gambar 5.6. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.6, *profit* kedua produk mengalami peningkatan. Untuk produk 1, *profit* mengalami peningkatan dari Rp51.128.125,00 sampai Rp80.416.000,00. Sedangkan untuk produk 2, peningkatan *profit* tidak terlalu signifikan, yaitu dari Rp642.490,00 sampai Rp3.538.750,00. Sedangkan pada gambar 5.7 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.7. dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp51.770.615,00 sampai Rp83.954.750,00. Parameter α yang digunakan pada produk 1 dan 2 adalah 9.000 sampai 13.000. Berdasarkan grafik pada gambar 5.7 dapat diketahui bahwa semakin tinggi nilai *potential customer* maka *total profit* akan semakin tinggi pula.



Gambar 5. 6 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Independen Parameter α



Gambar 5. 7 Grafik *Total Profit* Skenario 1 Produk Independen Parameter α

b. Produk Substitusi

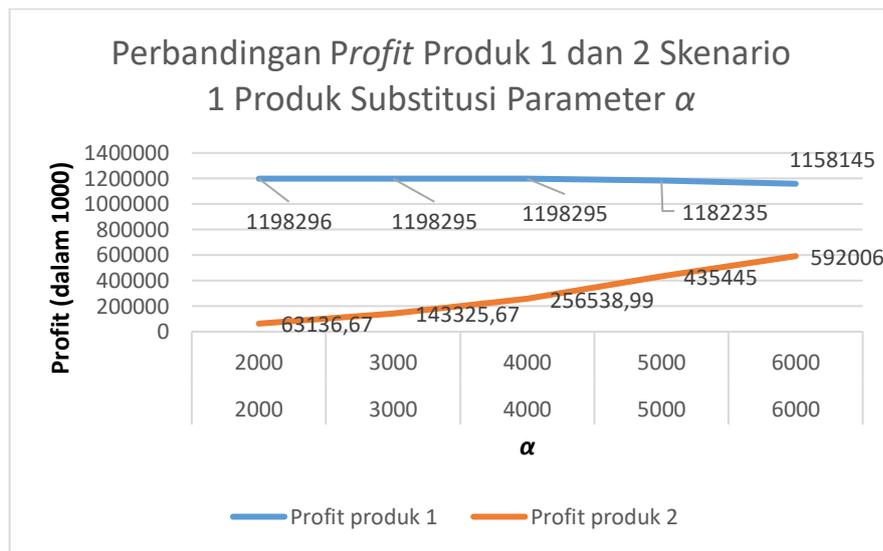
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 1.000 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.11.

Tabel 5. 11 *Output* Hasil Analisis Sensitivitas Parameter α

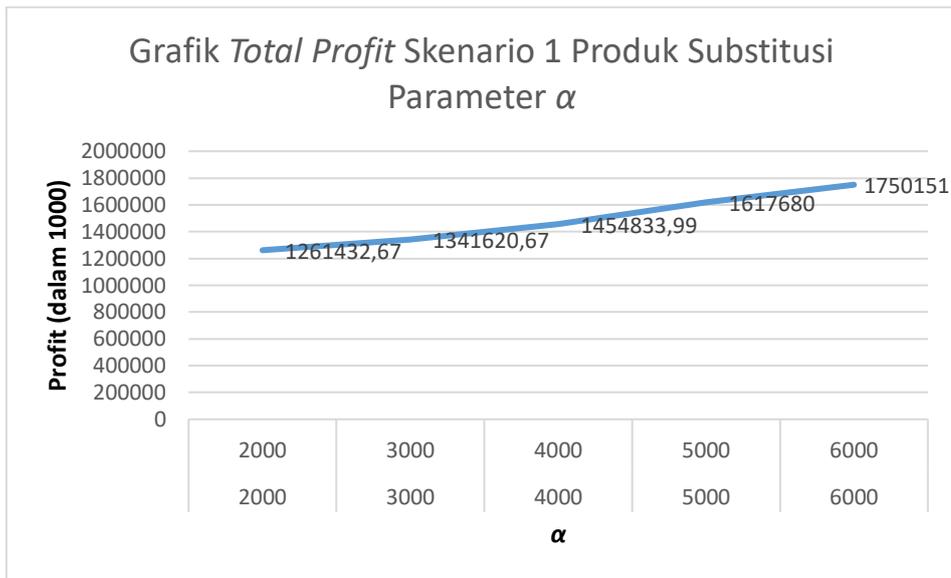
α										
Parameter		P (Rp1.000)		Q (unit)		T (hari)		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 1	produk 2	produk 1	produk 2	produk 1	
2.000	2.000	55	33	150	69	1	3	1.198.296	63.136,7	1.261.433
3.000	3.000	55	31	150	94	1	3	1.198.295	143.326	1.341.621
4.000	4.000	55	30	150	50	1	1	1.198.295	256.539	1.454.834
5.000	5.000	55	28	148	80	1	1	1.182.235	435.445	1.617.680
6.000	6.000	55	27	145	120	1	1	1.158.145	592.006	1.750.151

Grafik hasil analisis sensitivitas parameter α produk substitusi skenario 1 terlihat pada gambar 5.8 Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.8, *profit* produk 1 mengalami penurunan dari Rp1.198.296.000,00 sampai Rp1.158.145.000,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp63.136.670,00 sampai

Rp592.006.000,00. Sedangkan pada gambar 5.9 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.9 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp1.261.432.670,00 sampai Rp1.750.151.000,00. Parameter α yang digunakan pada produk 1 dan 2 adalah 2000 sampai 6000. Berdasarkan grafik pada gambar 5.9 dapat diketahui bahwa semakin tinggi nilai *potential customer* maka *total profit* akan semakin tinggi pula.



Gambar 5. 8 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter α



Gambar 5. 9 Grafik *Total Profit* Skenario 1 Produk Substitusi Parameter α

5.3.1.2 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

a. Produk Independen

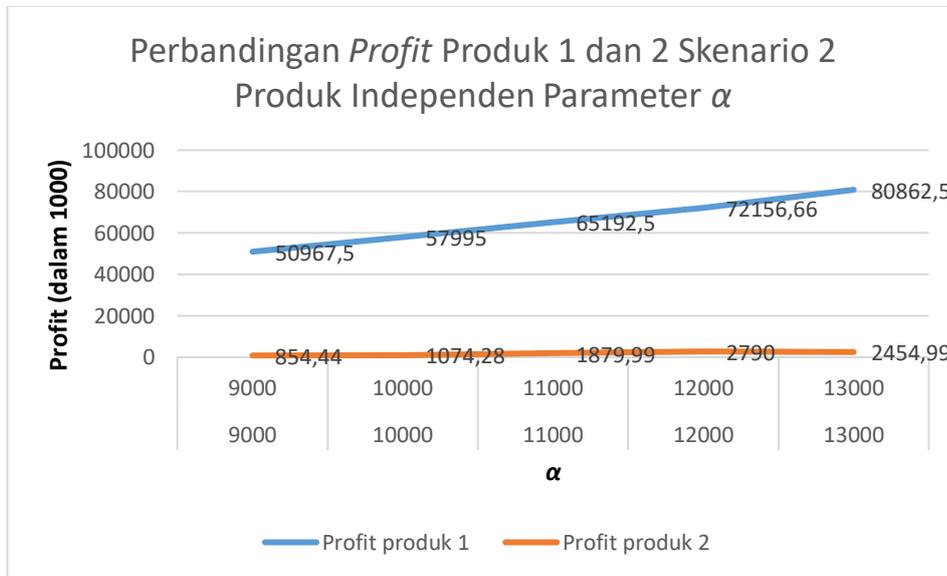
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 1.000 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.12.

Tabel 5. 12 *Output* Hasil Analisis Sensitivitas Parameter α

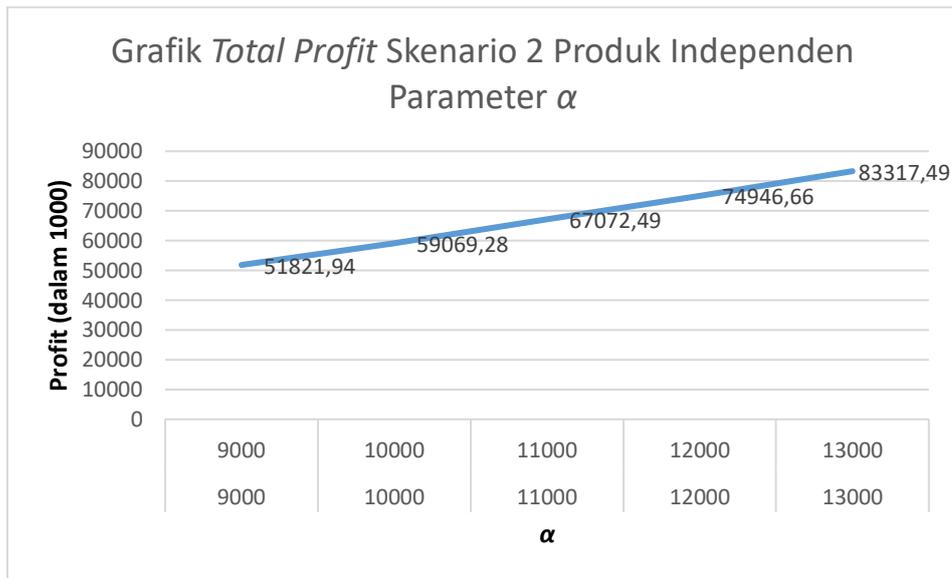
α										
Parameter		P (Rp1.000)		Q (unit)		T (hari)		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 1	produk 2	produk 1	produk 1	produk 2	produk 1	produk 2	produk 1	produk 1	
9.000	9.000	25	17	90	50	4	9	50967,5	854,44	51821,94
10.000	10.000	24	15	100	50	4	7	57995	1074,28	59069,28
11.000	11.000	24	17	91	51	3	9	65192,5	1879,99	67072,49
12.000	12.000	24	16	102	51	3	7	72156,66	2790	74946,66
13.000	13.000	24	17	80	51	2	9	80862,5	2454,99	83317,49

Grafik hasil analisis sensitivitas parameter α produk independen skenario 2 terlihat pada gambar 5.10. Gambar tersebut menggambarkan perbandingan *profit*

produk 1 dan 2. Berdasarkan grafik pada gambar 5.10, *profit* kedua produk mengalami peningkatan. Untuk produk 1, *profit* mengalami peningkatan dari Rp50.967.500,00 sampai Rp80.862.500,00. Sedangkan untuk produk 2, peningkatan *profit* tidak terlalu signifikan, yaitu dari Rp854.440,00 sampai Rp2.454.990,00. Sedangkan pada gambar 5.11 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.11 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp51.821.940,00 sampai Rp83.317.490,00. Parameter α yang digunakan pada produk 1 dan 2 adalah 9.000 sampai 13.000. Berdasarkan grafik pada gambar 5.11 dapat diketahui bahwa semakin tinggi nilai *potential customer* maka *total profit* akan semakin tinggi pula.



Gambar 5. 10 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter α



Gambar 5. 11 Grafik *Total Profit* Skenario 2 Produk Independen Parameter α

b. Produk Substitusi

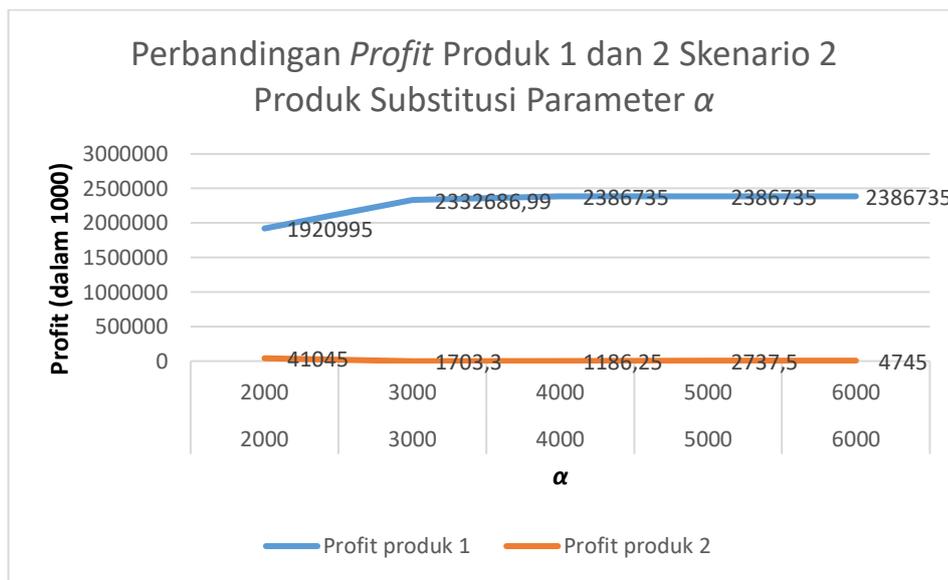
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 1.000 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.13.

Tabel 5. 13 *Output* Hasil Analisis Sensitivitas Parameter α

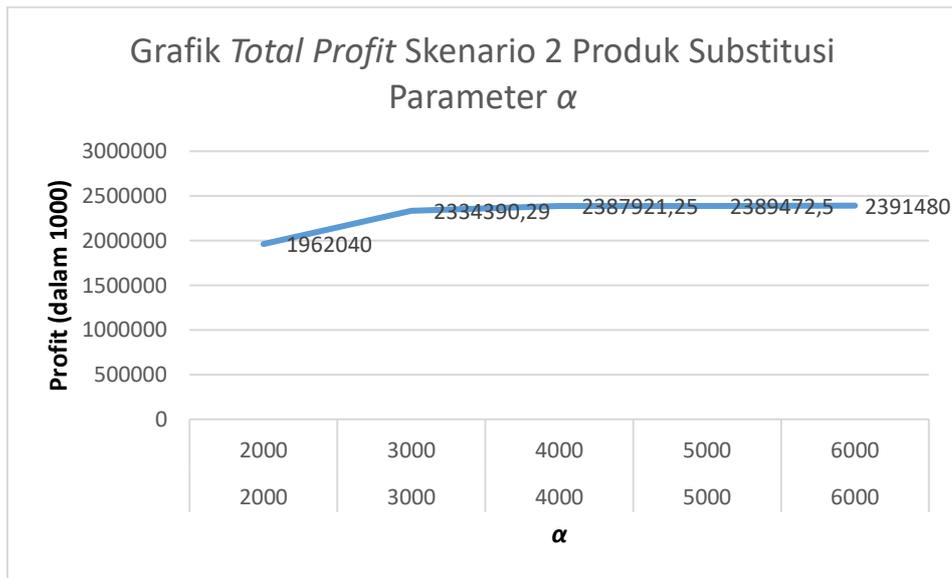
α										
Parameter		P (Rp1.000)		Q (unit)		T (hari)		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 1	produk 2	produk 1	produk 1	produk 2	produk 1	produk 2	produk 1	produk 1	
2.000	2.000	55	35	240	59	1	5	1.920.995	41.045	1.962.040
3.000	3.000	55	34	298	1	1	4	2.332.687	1.703,3	2.334.390
4.000	4.000	55	33	298	1	1	4	2.386.735	1.186,25	2.387.921
5.000	5.000	55	35	298	1	1	2	2.386.735	2.737,5	2.389.473
6.000	6.000	55	33	298	1	1	1	2.386.735	4.745	2.391.480

Grafik hasil analisis sensitivitas parameter α produk substitusi skenario 2 terlihat pada gambar 5.12. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.12, *profit* produk 1 mengalami peningkatan dari Rp1.920.995.000,00 sampai Rp2.386.735.000,00. Sedangkan

produk 2 mengalami penurunan *profit* dari Rp41.045.000,00 sampai Rp4.745.000,00. Sedangkan pada gambar 5.13 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.13 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp1.962.040.000,00 sampai Rp2.391.480.000,00. Parameter α yang digunakan pada produk 1 dan 2 adalah 2000 sampai 6000. Berdasarkan grafik pada gambar 5.13 dapat diketahui bahwa semakin tinggi nilai *potential customer* maka *total profit* akan semakin tinggi pula.



Gambar 5. 12 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter α



Gambar 5. 13 Grafik *Total Profit* Skenario 2 Produk Substitusi Parameter α

5.3.2 Analisis Sensitivitas Parameter β

Salah satu parameter yang digunakan dalam analisis sensitivitas adalah β (*shelf space efficiency of demand*). Analisis sensitivitas untuk parameter β dibagi ke dalam 2 skenario dengan 2 tipe produk. Analisis sensitivitas dilakukan dengan mengubah beberapa nilai pada parameter β terhadap produk yang diujikan. Parameter β diubah untuk mengetahui pengaruh parameter tersebut terhadap *total profit* pada 2 skenario. Berikut akan dibahas analisis sensitivitas untuk parameter β .

5.3.2.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

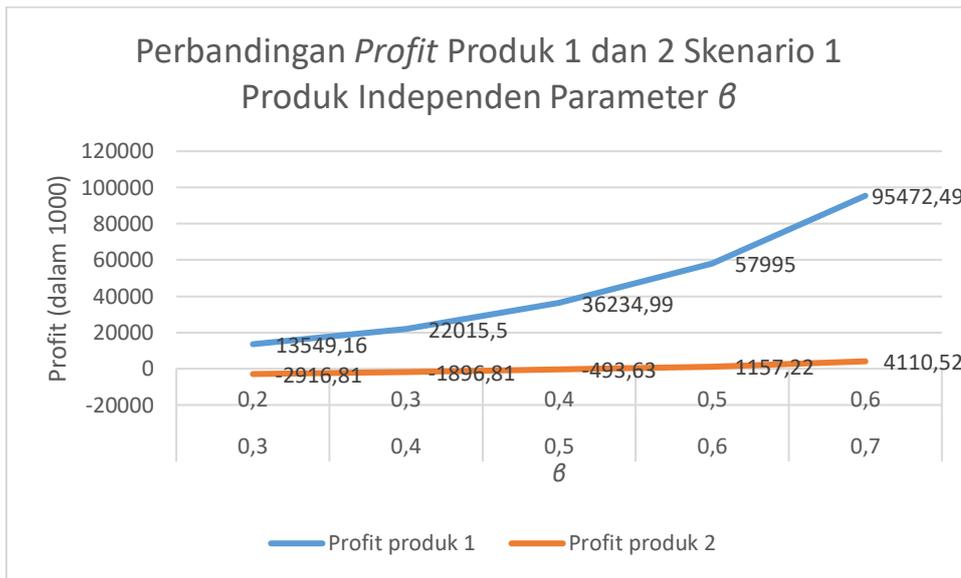
a. Produk Independen

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 0,1 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.14.

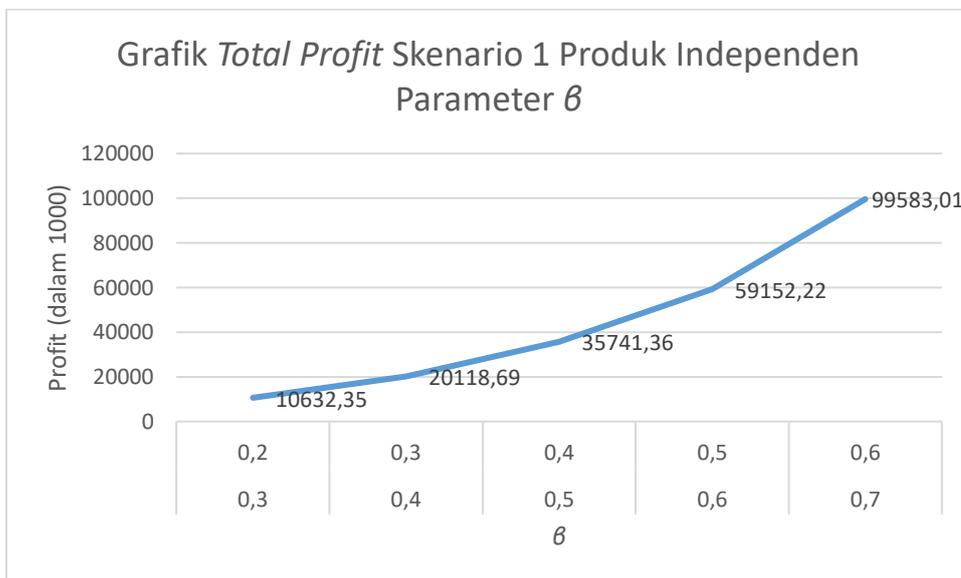
Tabel 5. 14 *Output* Hasil Analisis Sensitivitas Parameter β

β										
Parameter		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
0,3	0,2	24	15	50	50	6	11	13.549,16	-2.916,81	10.632,35
0,4	0,3	26	15	51	50	5	11	22.015,5	-1.896,81	20.118,69
0,5	0,4	25	16	78	51	5	11	36.234,99	-493,63	35.741,36
0,6	0,5	24	16	100	51	4	9	57.995	1.157,22	59.152,22
0,7	0,6	23	17	90	53	2	8	95.472,49	4.110,52	99.583,01

Grafik hasil analisis sensitivitas parameter β produk independen skenario 1 terlihat pada gambar 5.14. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.14. *profit* produk 1 mengalami peningkatan dari Rp13.549.160,00 sampai Rp95.472.490,00. Sedangkan produk 2 juga mengalami peningkatan *profit*, yaitu dari -Rp2.916.810,00 sampai Rp4.110.520,00. Sedangkan pada gambar 5.15 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.15 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp10.632.350,00 sampai Rp99.583.010,00. Parameter β yang digunakan pada produk 1 adalah 0,3 sampai 0,7 dan pada produk 2 adalah 0,2 sampai 0,6. Berdasarkan grafik pada gambar 5.15 dapat diketahui bahwa semakin tinggi nilai *shelf space efficiency of demand* maka *total profit* akan semakin tinggi pula.



Gambar 5. 14 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Independen Parameter β



Gambar 5. 15 Grafik *Total Profit* Skenario 1 Produk Independen Parameter β

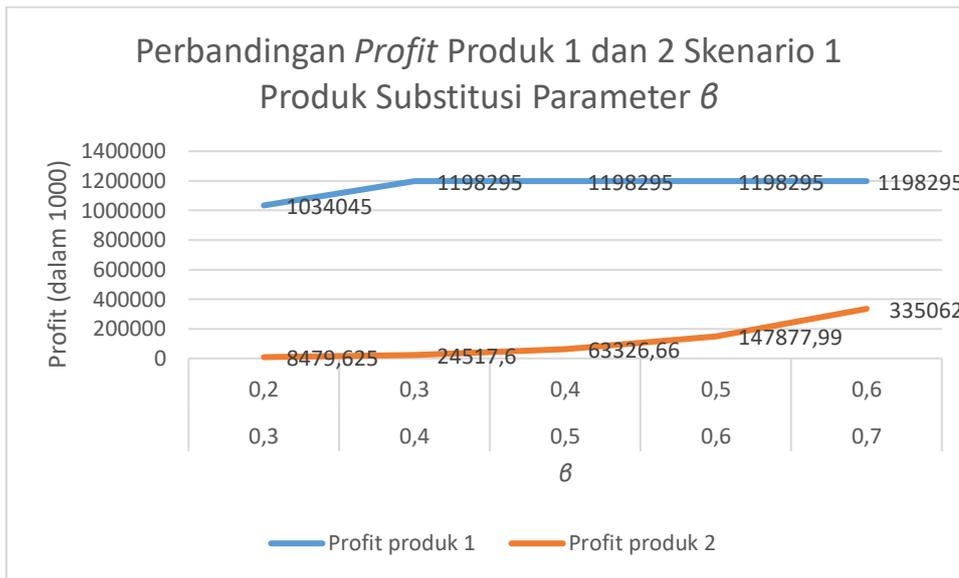
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 0,1 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.15.

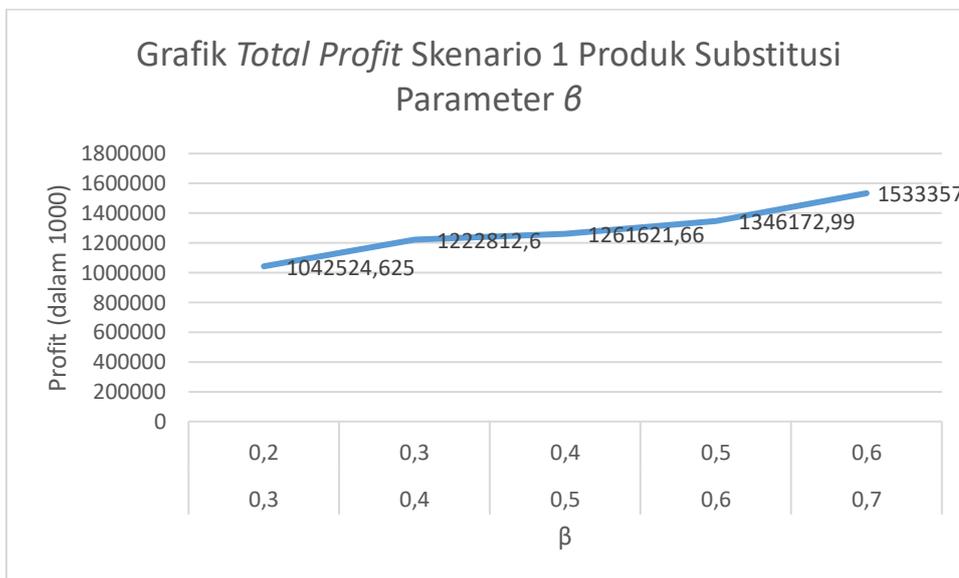
Tabel 5. 15 *Output* Hasil Analisis Sensitivitas Parameter β

β										
Parameter		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
0,3	0,2	52	35	150	50	1	8	1.034.045	8.479,625	1.042.525
0,4	0,3	55	35	150	51	1	5	1.198.295	24.517,6	1.222.813
0,5	0,4	55	33	150	76	1	3	1.198.295	63.326,66	1.261.622
0,6	0,5	55	31	150	77	1	2	1.198.295	147.877,99	1.346.173
0,7	0,6	55	29	150	59	1	1	1.198.295	335.062	1.533.357

Grafik hasil analisis sensitivitas parameter β produk substitusi skenario 1 terlihat pada gambar 5.16. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.16. *Profit* produk 1 tidak mengalami peningkatan maupun penurunan. Nilai *profit* datar, yaitu Rp432.252.000,00. Sedangkan produk 2 mengalami peningkatab *profit* dari Rp1.704.750.000,00 sampai Rp124.465.000,00. Sedangkan pada gambar 5.17 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.17 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp434.229.750,00 sampai Rp556.990.000,00. Parameter β yang digunakan pada produk 1 adalah 0,3 sampai 0,7 dan pada produk 2 adalah 0,2 sampai 0,6. Berdasarkan grafik pada gambar 5.17 dapat diketahui bahwa semakin tinggi nilai *shelf space efficiency of demand* maka *total profit* akan semakin tinggi pula.



Gambar 5. 16 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter β (dalam 1000)



Gambar 5. 17 Grafik *Total Profit* Skenario 1 Produk Substitusi Parameter β (dalam 1000)

5.3.2.2 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

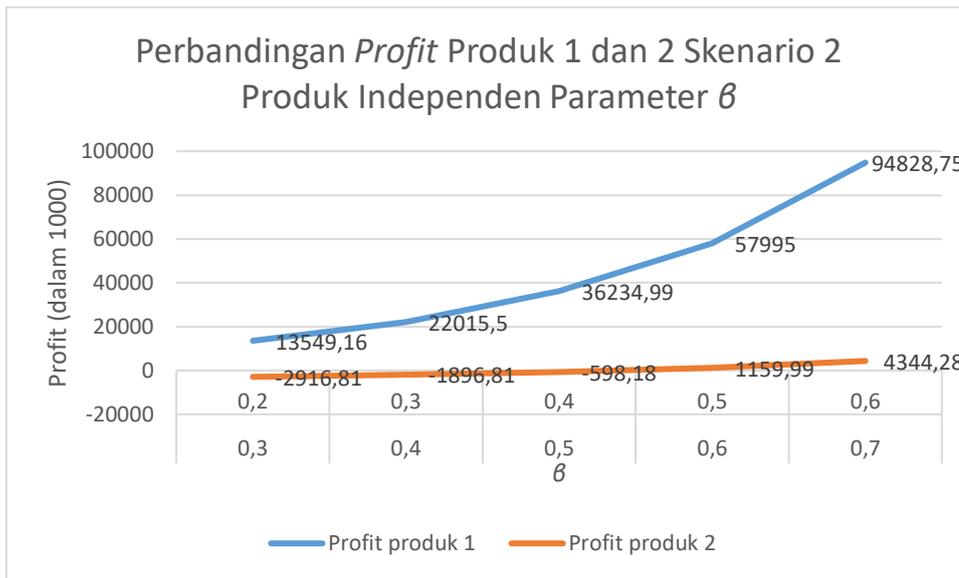
- a. Produk Independen

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 0,1 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.16.

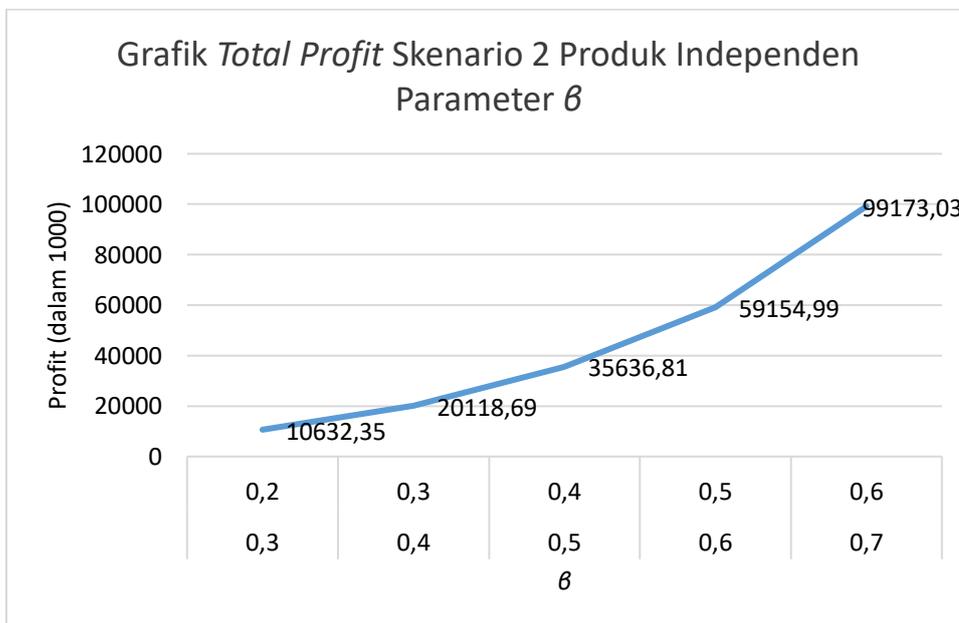
Tabel 5. 16 *Output* Hasil Analisis Sensitivitas Parameter β

β										
Parameter		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
0,3	0,2	24	15	50	50	6	11	13.549,16	-2.916,81	10.632,35
0,4	0,3	26	15	51	50	5	11	22.015,5	-1.896,81	20.118,69
0,5	0,4	25	16	78	52	5	11	36.234,99	-598,18	35.636,81
0,6	0,5	24	15	100	50	4	8	57.995	1.159,99	59.154,99
0,7	0,6	23	17	91	53	2	7	94.828,75	4.344,28	99.173,03

Grafik hasil analisis sensitivitas parameter β produk independen skenario 2 terlihat pada gambar 5.18. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.18. *Profit* produk 1 mengalami peningkatan *profit*, yaitu dari Rp13.549.160,00 sampai Rp94.828.750,00. Sedangkan produk 2 juga mengalami peningkatan dari -Rp2.916.810,00 sampai Rp4.344.280,00. Sedangkan pada gambar 5.19 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.19 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp10.632.350,00 sampai Rp99.173.030,00. Parameter β yang digunakan pada produk 1 adalah 0,3 sampai 0,7 dan pada produk 2 adalah 0,2 sampai 0,6. Berdasarkan grafik pada gambar 5.19 dapat diketahui bahwa semakin tinggi nilai *shelf space efficiency of demand* maka *total profit* akan semakin tinggi pula.



Gambar 5. 18 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter β



Gambar 5. 19 Grafik *Total Profit* Skenario 2 Produk Independen Parameter β

b. Produk Substitusi

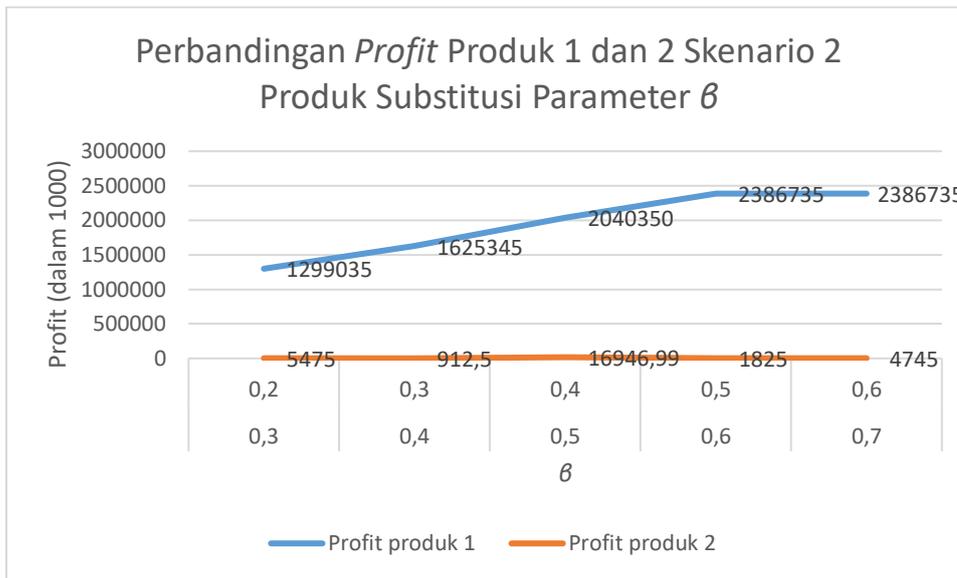
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan

dengan skala 0,1 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.17.

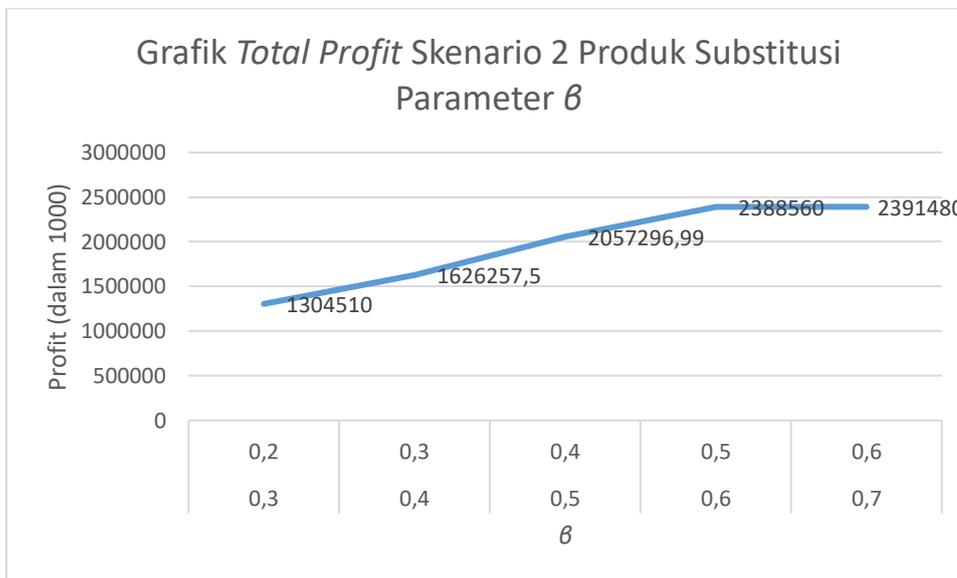
Tabel 5. 17 *Output* Hasil Analisis Sensitivitas Parameter β

β										
Parameter		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
0,3	0,2	45	35	298	1	1	1	1.299.035	5475	1.304.510
0,4	0,3	48	35	298	1	1	6	1.625.345	912,5	1.626.258
0,5	0,4	54	35	267	32	1	1	2.040.350	16.946,99	2.057.297
0,6	0,5	55	35	298	1	1	3	2.386.735	1.825	2.388.560
0,7	0,6	55	33	298	1	1	1	2.386.735	4.745	2.391.480

Grafik hasil analisis sensitivitas parameter β produk substitusi skenario 2 terlihat pada gambar 5.20. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.20. *Profit* produk 1 mengalami peningkatan dari Rp1.203.749.000,00 sampai Rp2.386.735.000,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp123.241.000,00 sampai Rp178.650.000,00. Kemudian mengalami penurunan sampai Rp1.186.000,00. Sedangkan pada gambar 5.21 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.21 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp1.326.990.000,00 sampai Rp2.387.921.250,00. Parameter β yang digunakan pada produk 1 adalah 0,3 sampai 0,7 dan pada produk 2 adalah 0,2 sampai 0,6. Berdasarkan grafik pada gambar 5.21 dapat diketahui bahwa semakin tinggi nilai *shelf space efficiency of demand* maka *total profit* akan semakin tinggi pula.



Gambar 5. 20 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter β



Gambar 5. 21 Grafik *Total Profit* Skenario 2 Produk Substitusi Parameter β

5.3.3 Analisis Sensitivitas Parameter λ

Salah satu parameter yang digunakan dalam analisis sensitivitas adalah λ (price efficiency of demand). Analisis sensitivitas untuk parameter λ dibagi ke dalam 2 skenario dengan 2 tipe produk. Analisis sensitivitas dilakukan dengan mengubah beberapa nilai pada parameter λ terhadap produk yang diujikan. Parameter λ diubah untuk mengetahui pengaruh parameter tersebut terhadap *total*

profit pada 2 skenario. Berikut akan dibahas analisis sensitivitas untuk parameter λ .

5.3.3.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

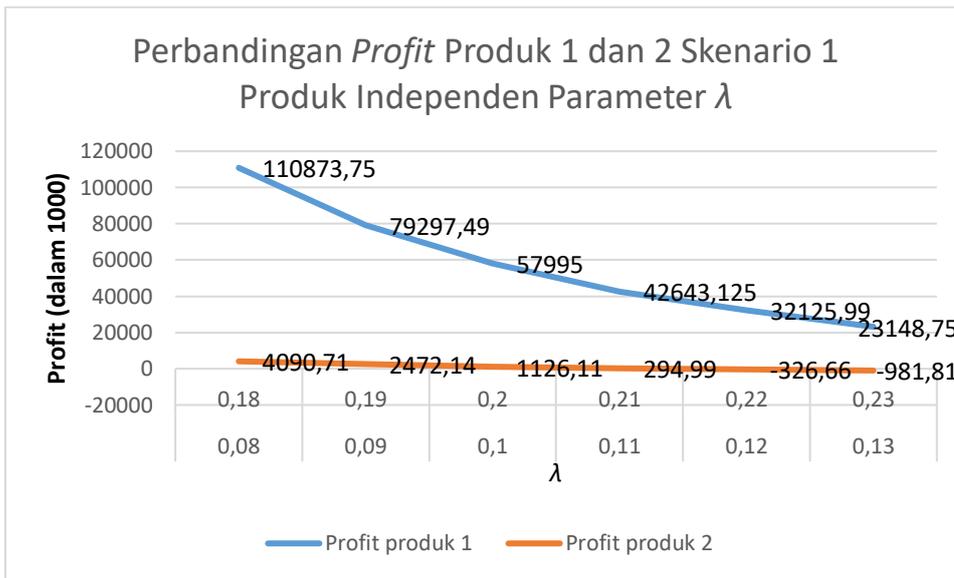
a. Produk Independen

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 0,01 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.18.

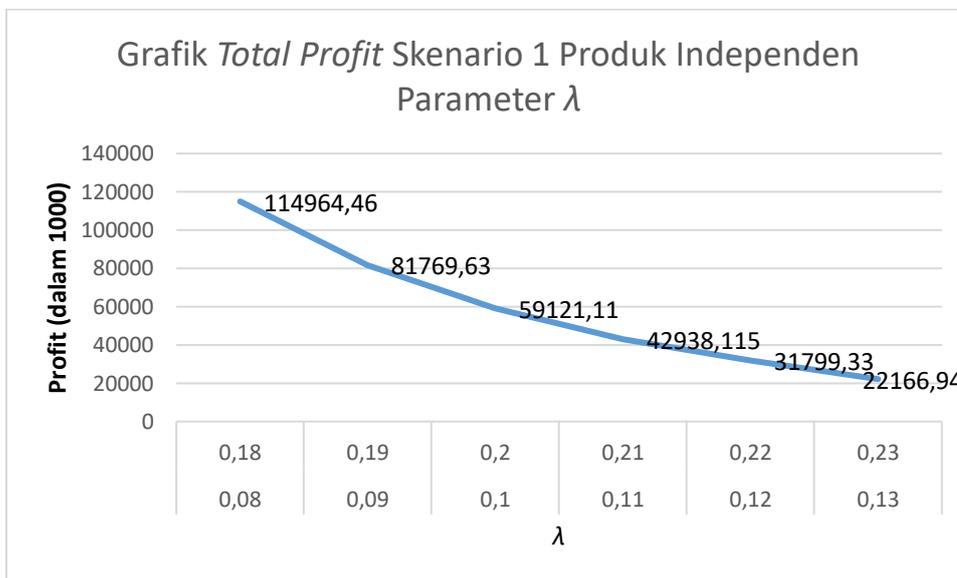
Tabel 5. 18 *Output* Hasil Analisis Sensitivitas Parameter λ

λ										
Parameter		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
0,08	0,18	27	17	81	50	2	7	110.873,8	4.090,71	114.964,46
0,09	0,19	26	16	91	51	3	7	79.297,49	2.472,14	81.769,63
0,1	0,2	24	18	100	51	4	9	57.995	1.126,11	59.121,11
0,11	0,21	24	16	83	50	4	9	42.643,13	294,99	42.938,115
0,12	0,22	23	16	88	51	5	9	32.125,99	-326,66	31.799,33
0,13	0,23	23	15	89	51	6	11	23.148,75	-981,81	22.166,94

Grafik hasil analisis sensitivitas parameter λ produk independen skenario 1 terlihat pada gambar 5.22. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.22. *Profit* produk 1 mengalami penurunan dari Rp110.873.750,00 sampai Rp23.148.750,00. Sedangkan produk 2 mengalami penurunan dari Rp4.090.710,00 sampai -Rp981.810,00. Sedangkan pada gambar 5.23 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.23 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp114.964.460,00 sampai Rp22.166.940,00. Parameter λ yang digunakan pada produk 1 adalah 0,08 sampai 0,13 dan pada produk 2 adalah 0,18 sampai 0,23. Berdasarkan grafik pada gambar 5.23 dapat diketahui bahwa semakin tinggi nilai *price efficiency of demand* maka *total profit* akan semakin kecil.



Gambar 5. 22 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Independen Parameter λ



Gambar 5. 23 Grafik *Total Profit* Skenario 1 Produk Independen Parameter λ

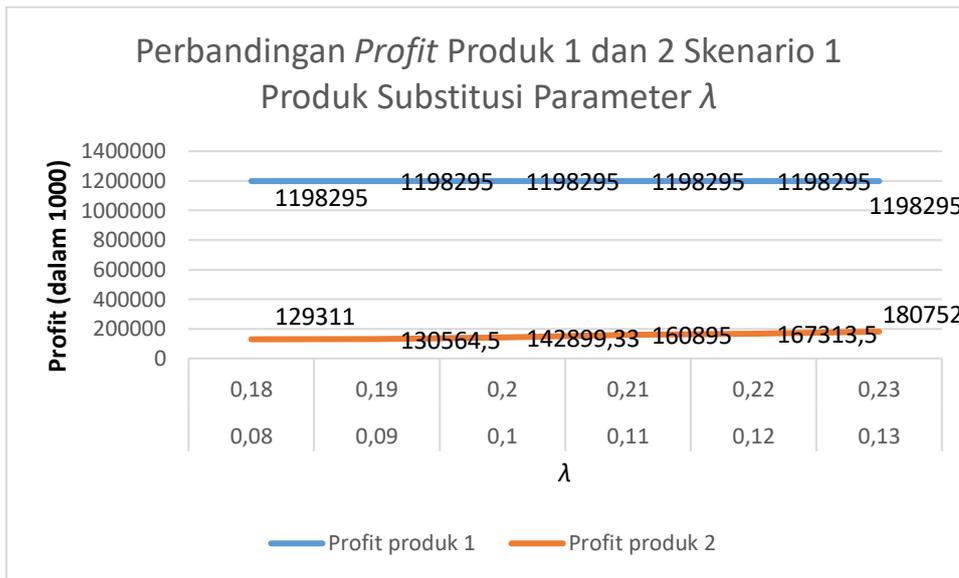
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 0,01 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.19.

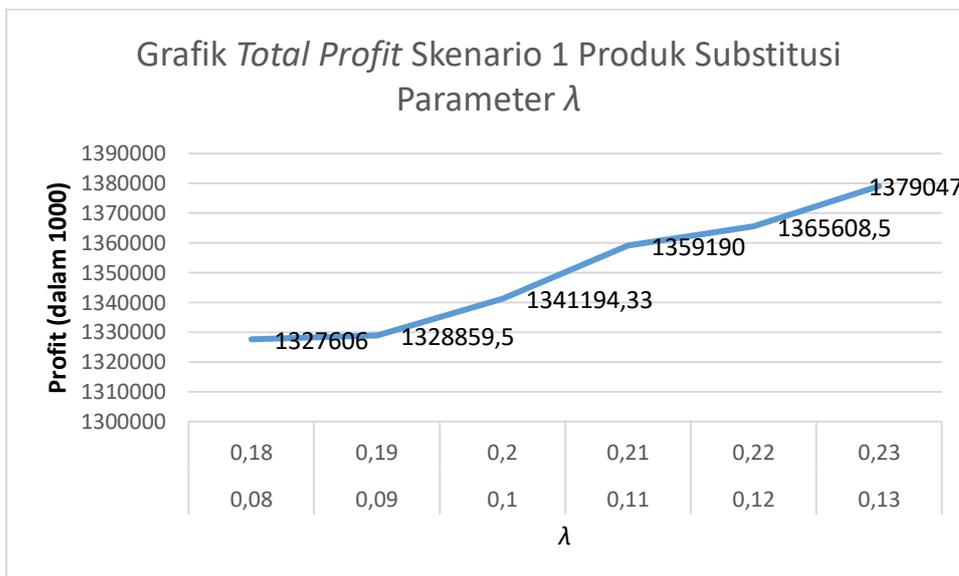
Tabel 5. 19 *Output* Hasil Analisis Sensitivitas Parameter λ

λ										
Parameter		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp.1000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
0,08	0,18	55	28	150	73	1	2	1.198.295	129.311	1.327.606
0,09	0,19	55	30	150	72	1	2	1.198.295	130.564,5	1.328.860
0,1	0,2	55	31	150	102	1	3	1.198.295	142.899,33	1.341.194
0,11	0,21	55	32	150	70	1	2	1.198.295	160.895	1.359.190
0,12	0,22	55	33	150	90	1	2	1.198.295	167.313,5	1.365.609
0,13	0,23	55	34	150	75	1	2	1.198.295	180.752	1.379.047

Grafik hasil analisis sensitivitas parameter λ produk substitusi skenario 1 terlihat pada gambar 5.24. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.24. *Profit* produk 1 mengalami tidak mengalami peningkatan maupun penurunan, yaitu datar diangka Rp1.198.295.000,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp129.311.000,00 sampai Rp180.752.000,00. Sedangkan pada gambar 5.25 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.25 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp1.327.606.000,00 sampai Rp1.379.047.000,00. Parameter λ yang digunakan pada produk 1 adalah 0,08 sampai 0,13 dan pada produk 2 adalah 0,18 sampai 0,23. Berdasarkan grafik pada gambar 5.25 dapat diketahui bahwa semakin tinggi nilai *price efficiency of demand* maka *total profit* akan semakin tinggi pula.



Gambar 5. 24 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter λ



Gambar 5. 25 Grafik *Total Profit* Skenario 1 Produk Substitusi Parameter λ

5.3.3.2 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

a. Produk Independen

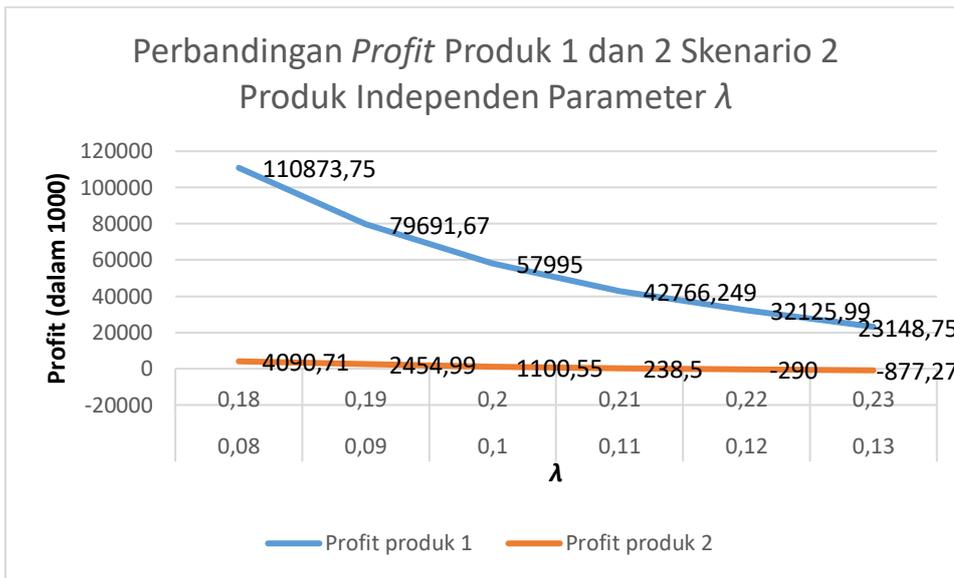
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan

dengan skala 0,01 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.20.

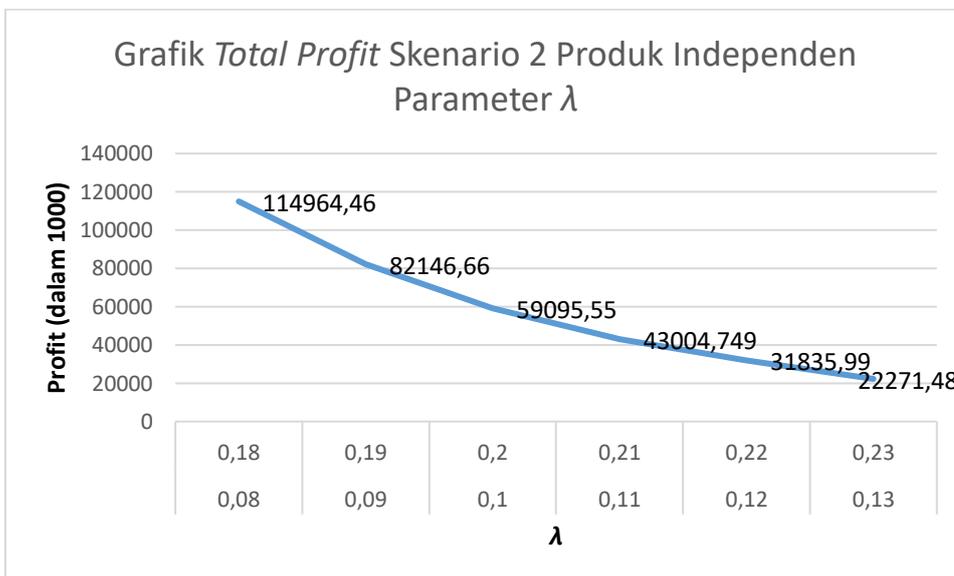
Tabel 5. 20 *Output* Hasil Analisis Sensitivitas Parameter λ

λ										
Parameter		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
0,08	0,18	27	17	81	50	2	7	110.873,8	4.090,71	114.964,5
0,09	0,19	26	17	98	51	3	9	79.691,67	2.454,99	82.146,66
0,1	0,2	24	15	100	51	4	9	57.995	1.100,55	59.095,55
0,11	0,21	24	17	86	50	4	10	42.766,25	238,5	43.004,75
0,12	0,22	23	16	88	51	5	11	32.125,99	-290	31.835,99
0,13	0,23	23	15	89	50	6	11	23.148,75	-877,27	22.271,48

Grafik hasil analisis sensitivitas parameter λ produk independen skenario 2 terlihat pada gambar 5.26. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.26. *Profit* produk 1 mengalami penurunan dari Rp110.873.750,00 sampai Rp23.148.750,00. Sedangkan produk 2 mengalami penurunan dari Rp4.090.710,00 sampai -Rp981.810,00. Sedangkan pada gambar 5.27 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.27 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp114.964.460,00 sampai Rp22.166.940,00. Parameter λ yang digunakan pada produk 1 adalah 0,08 sampai 0,13 dan pada produk 2 adalah 0,18 sampai 0,23. Berdasarkan grafik pada gambar 5.27 dapat diketahui bahwa semakin tinggi nilai *price efficiency of demand* maka *total profit* akan semakin kecil.



Gambar 5. 26 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter λ



Gambar 5. 27 Grafik *Total Profit* Skenario 2 Produk Independen Parameter λ

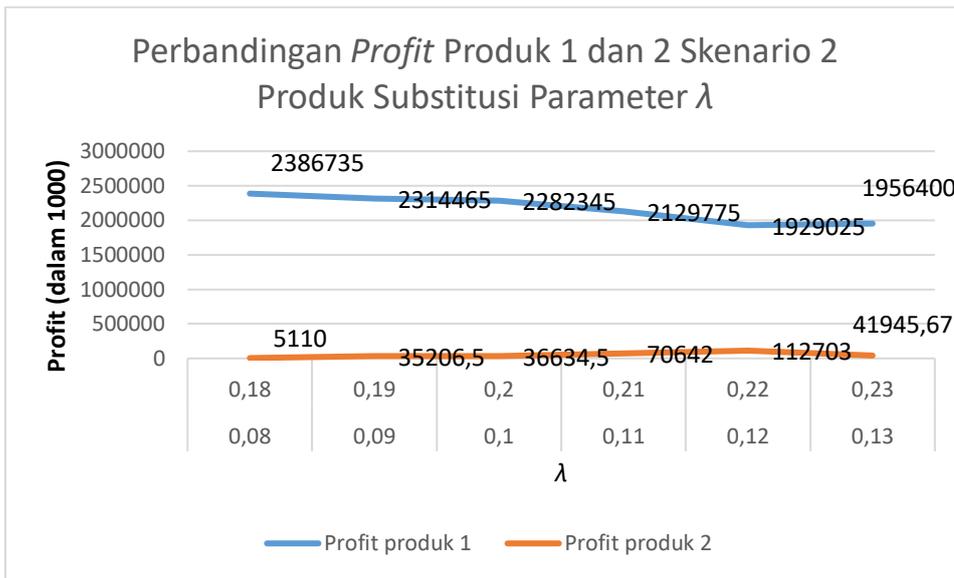
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 0,01 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.20.

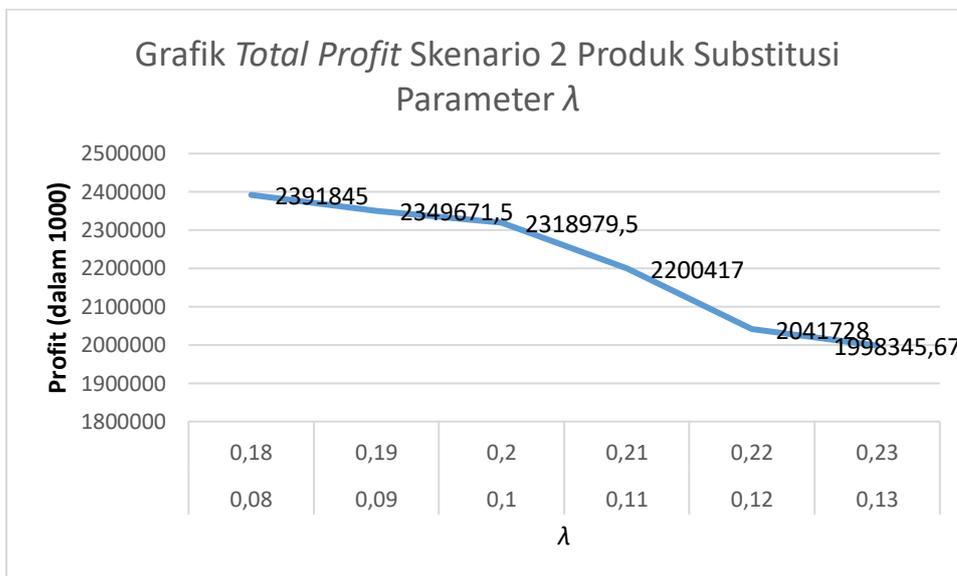
Tabel 5. 21 *Output* Hasil Analisis Sensitivitas Parameter λ

λ										
Parameter		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
0,08	0,18	55	34	298	1	1	1	2.386.735	5.110	2.391.845
0,09	0,19	55	33	289	10	1	2	2.314.465	35.206,5	2.349.672
0,1	0,2	55	35	285	14	1	2	2.282.345	36.634,5	2.318.980
0,11	0,21	55	35	266	33	1	1	2.129.775	70.642	2.200.417
0,12	0,22	55	35	241	58	1	2	1.929.025	112.703	2.041.728
0,13	0,23	52	35	283	16	1	3	1.956.400	41.945,67	1.998.346

Grafik hasil analisis sensitivitas parameter λ produk substitusi skenario 2 terlihat pada gambar 5.28. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.28. *Profit* produk 1 mengalami penurunan dari Rp2.386.735.000,00 sampai Rp1.929.025.000,00. Kemudian mengalami peningkatan sampai Rp1.956.400.000,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp5.110.000,00 sampai Rp112.703.000,00. Kemudian mengalami penurunan sampai Rp41.945.670,00. Sedangkan pada gambar 5.29 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.29 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp2.391.845.000,00 sampai Rp1.998.345.670,00. Parameter λ yang digunakan pada produk 1 adalah 0,08 sampai 0,13 dan pada produk 2 adalah 0,18 sampai 0,23. Berdasarkan grafik pada gambar 5.29 dapat diketahui bahwa semakin tinggi nilai *price efficiency of demand* maka *total profit* akan semakin kecil.



Gambar 5. 28 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter λ



Gambar 5. 29 Grafik *Total Profit* Skenario 2 Produk Substitusi Parameter λ

5.3.4 Analisis Sensitivitas Parameter c

Salah satu parameter yang digunakan dalam analisis sensitivitas adalah c (purchasing cost). Analisis sensitivitas untuk parameter c dibagi ke dalam 2 skenario dengan 2 tipe produk. Analisis sensitivitas dilakukan dengan mengubah beberapa nilai pada parameter c terhadap produk yang diujikan. Parameter c

diubah untuk mengetahui pengaruh parameter tersebut terhadap *total profit* pada 2 skenario. Berikut akan dibahas analisis sensitivitas untuk parameter *c*.

5.3.4.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

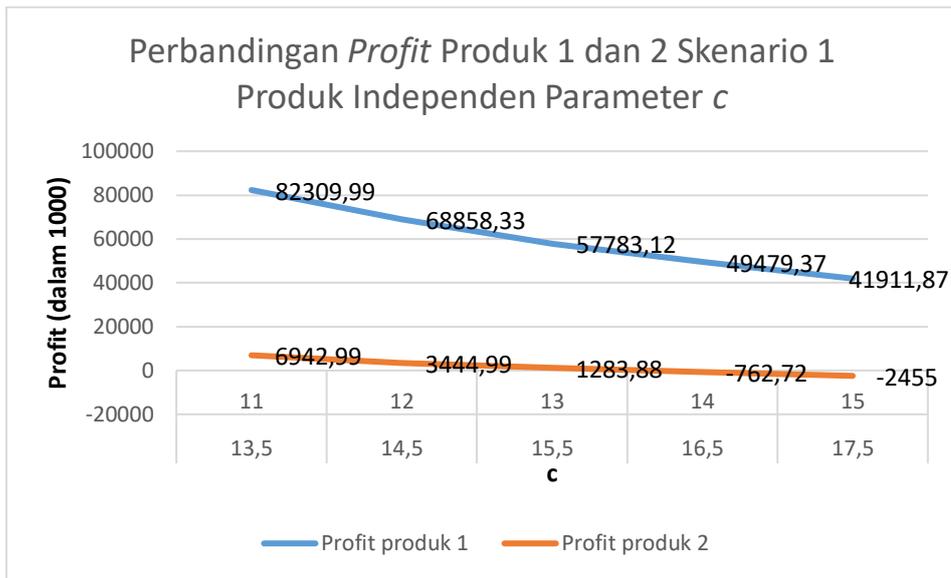
a. Produk Independen

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.22.

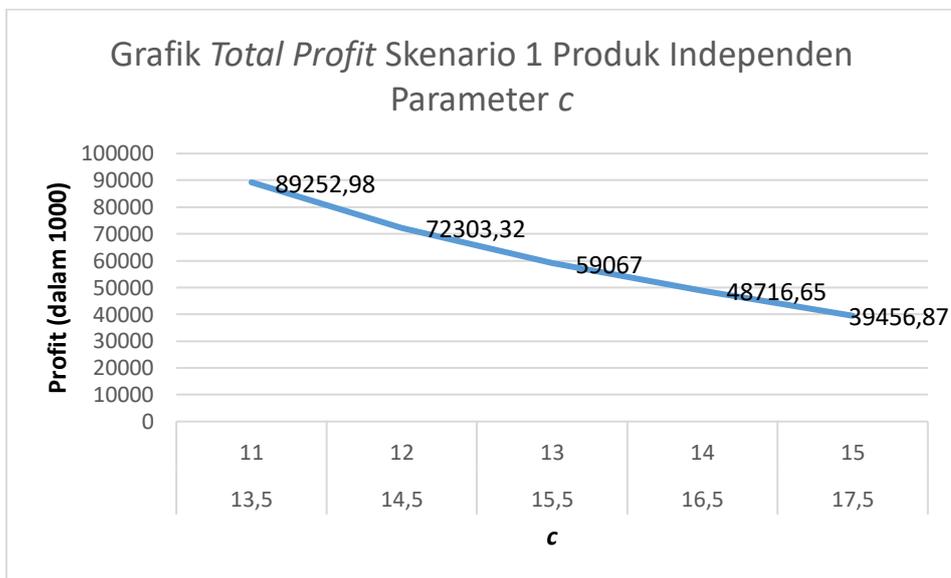
Tabel 5. 22 *Output* Hasil Analisis Sensitivitas Parameter *c*

<i>c</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
13,5	11	22	15	80	65	2	5	82.309,99	6.942,99	89.252,98
14,5	12	24	15	96	52	3	6	68.858,33	3.444,99	72.303,32
15,5	13	25	16	91	50	4	9	57.783,12	1.283,88	59.067
16,5	14	25	16	91	51	4	11	49.479,37	-762,72	48.716,65
17,5	15	27	16	81	51	4	11	41.911,87	-2.455	39.456,87

Grafik hasil analisis sensitivitas parameter *c* produk independen skenario 1 terlihat pada gambar 5.30. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.30. *Profit* produk 1 mengalami penurunan dari Rp82.309.990,00 sampai Rp41.911.870,00. Sedangkan produk 2 mengalami penurunan dari Rp6.942.990,00 sampai -Rp2.455.000,00. Sedangkan pada gambar 5.31 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.31 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp89.252.980,00 sampai Rp39.456.870,00. Parameter *c* yang digunakan pada produk 1 adalah 13,5 sampai 17,5 dan pada produk 2 adalah 11 sampai 15. Berdasarkan grafik pada gambar 5.31 dapat diketahui bahwa semakin besar nilai *purchasing cost* maka *total profit* akan semakin kecil.



Gambar 5. 30 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Independen Parameter *c*



Gambar 5. 31 Grafik *Total Profit* Skenario 1 Produk Independen Parameter *c*

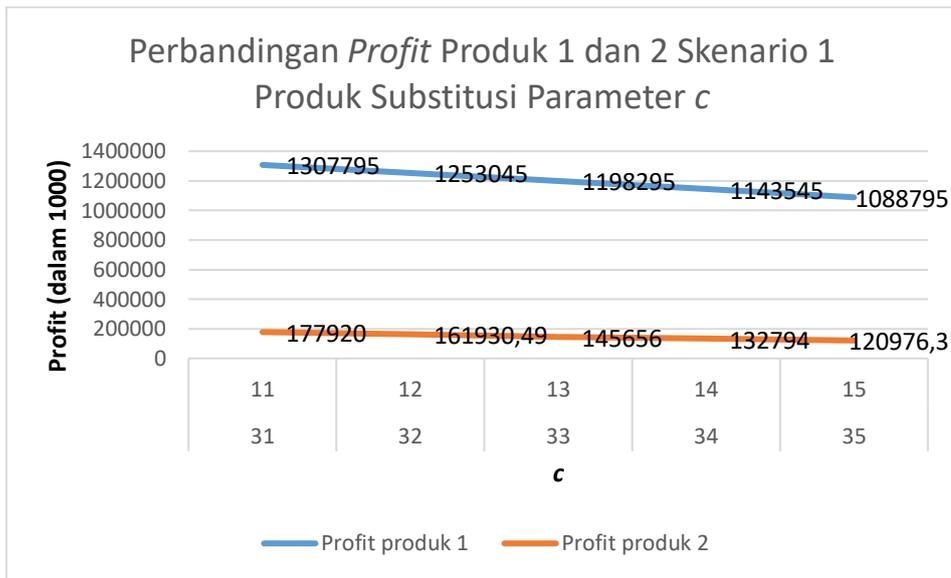
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.23.

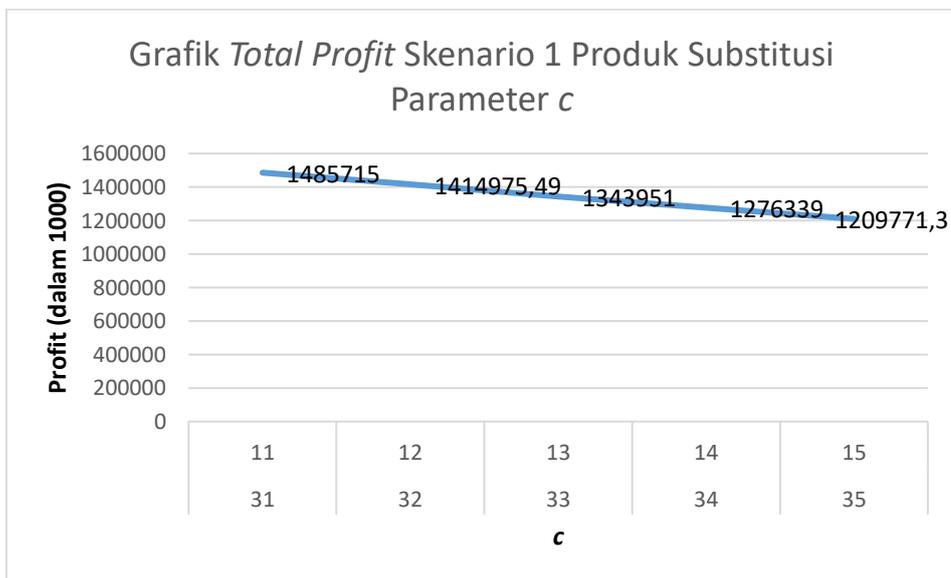
Tabel 5. 23 *Output* Hasil Analisis Sensitivitas Parameter *c*

<i>c</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
31	11	55	31	150	51	1	1	1.307.795	177.920	1.485.715
32	12	55	31	150	77	1	2	1.253.045	161.930,49	1.414.975
33	13	55	31	150	81	1	2	1.198.295	145.656	1.343.951
34	14	55	31	150	68	1	2	1.143.545	132.794	1.276.339
35	15	55	31	150	89	1	3	1.088.795	120.976,3	1.209.771

Grafik hasil analisis sensitivitas parameter *c* produk substitusi skenario 1 terlihat pada gambar 5.32. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.32. *Profit* produk 1 mengalami penurunan dari Rp1.307.795.000,00 sampai Rp1.929.025.000,00. Kemudian mengalami peningkatan sampai Rp1.088.795.000,00. Sedangkan produk 2 juga mengalami penurunan *profit* dari Rp177.920.000,00 sampai Rp120.976.300,00. Sedangkan pada gambar 5.33 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.33 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp1.485.715.000,00 sampai Rp1.209.771.300,00. Parameter *c* yang digunakan pada produk 1 adalah 31 sampai 35 dan pada produk 2 adalah 11 sampai 15. Berdasarkan grafik pada gambar 5.33 dapat diketahui bahwa semakin besar nilai *purchasing cost* maka *total profit* akan semakin kecil.



Gambar 5. 32 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter *c*



Gambar 5. 33 Grafik *Total Profit* Skenario 1 Produk Substitusi Parameter *c*

5.3.4.2 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

a. Produk Independen

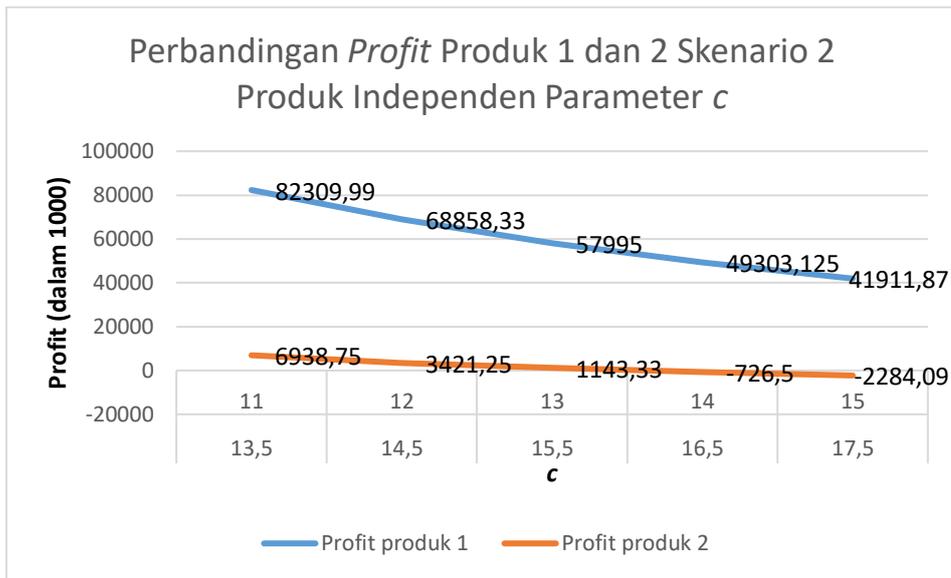
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan

dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.24.

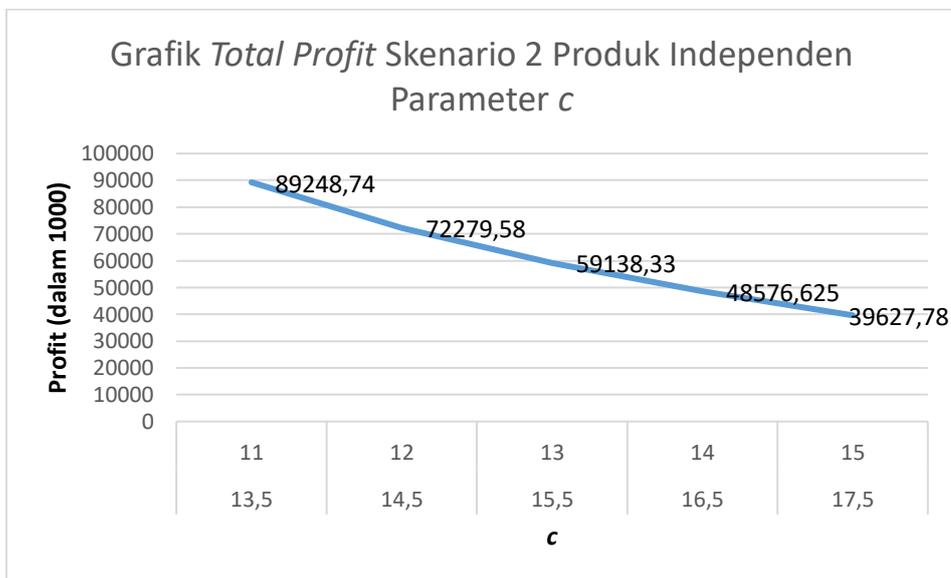
Tabel 5. 24 *Output* Hasil Analisis Sensitivitas Parameter *c*

<i>c</i>										
Parameter (Rp1.000)		P		Q		T		Profit		Total Profit
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
13,5	11	22	15	80	55	2	4	82.309,99	6.938,75	89.248,74
14,5	12	24	15	96	55	3	8	68.858,33	3.421,25	72.279,58
15,5	13	24	17	100	50	4	9	57.995	1.143,33	59.138,33
16,5	14	27	15	81	50	4	10	49.303,13	-726,5	48.576,63
17,5	15	27	16	81	50	4	11	41.911,87	-2.284,09	39.627,78

Grafik hasil analisis sensitivitas parameter *c* produk substitusi skenario 2 terlihat pada gambar 5.34. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.34. *Profit* produk 1 mengalami penurunan dari Rp82.309.990,00 sampai Rp41.911.870,00. Sedangkan produk 2 juga mengalami penurunan *profit* dari Rp6.938.750,00 sampai -Rp2.284.090,00. Sedangkan pada gambar 5.35 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.35 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp89.248.740,00 sampai Rp39.627.780,00. Parameter *c* yang digunakan pada produk 1 adalah 13,5 sampai 17,5 dan pada produk 2 adalah 11 sampai 15. Berdasarkan grafik pada gambar 5.35 dapat diketahui bahwa semakin besar nilai *purchasing cost* maka *total profit* akan semakin kecil.



Gambar 5. 34 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter *c*



Gambar 5. 35 Grafik *Total Profit* Skenario 2 Produk Independen Parameter *c*

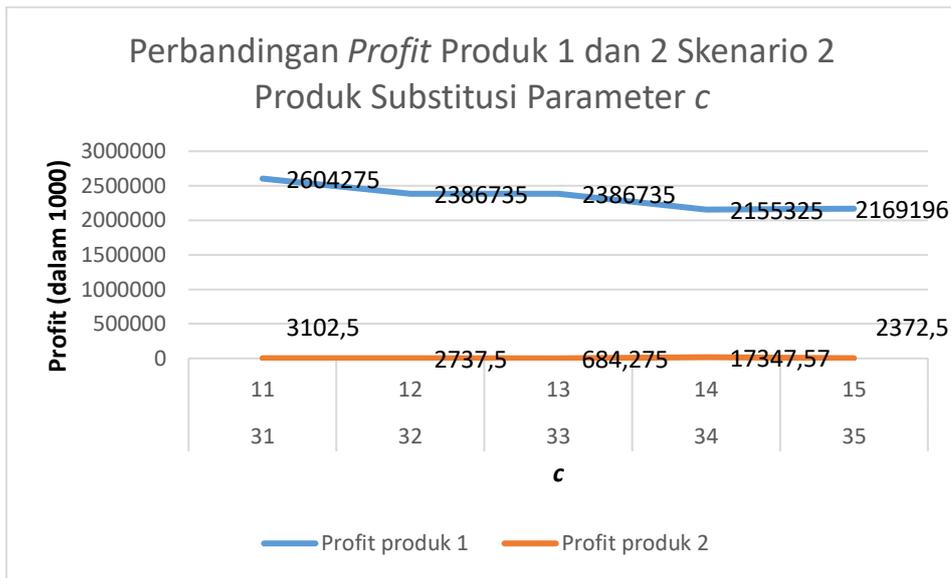
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.25.

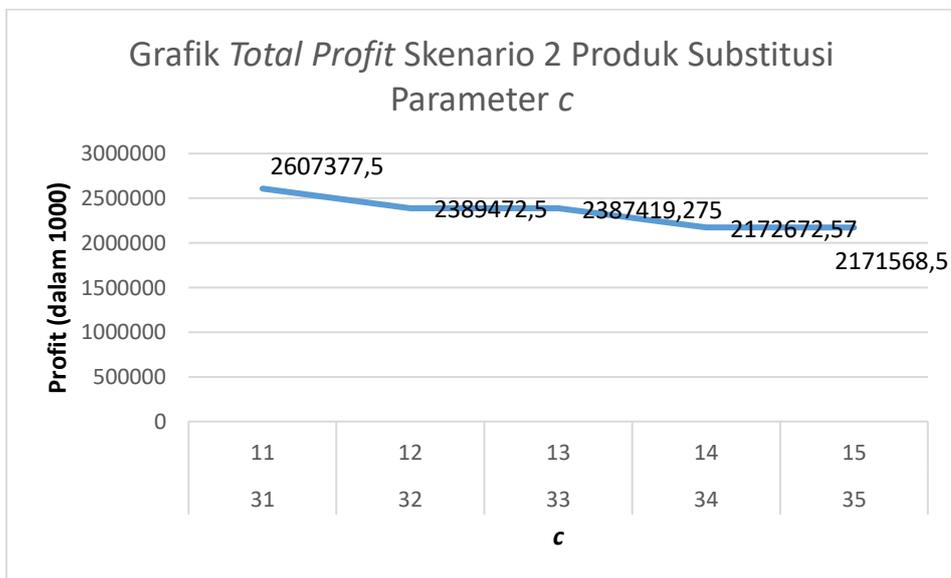
Tabel 5. 25 *Output* Hasil Analisis Sensitivitas Parameter *c*

<i>c</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
31	11	55	35	298	1	1	2	2.604.275	3.102,5	2.607.378
32	12	54	34	298	1	1	2	2.386.735	2737,5	2.389.473
33	13	55	35	298	1	1	8	2.386.735	684,275	2.387.419
34	14	55	34	282	17	1	7	2.155.325	17.347,57	2.172.673
35	15	55	35	298	1	1	2	2.169.196	2.372,5	2.171.569

Grafik hasil analisis sensitivitas parameter *c* produk substitusi skenario 2 terlihat pada gambar 5.36. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.36. *Profit* produk 1 mengalami penurunan dari Rp2.604.275.000,00 sampai Rp2.169.196.000,00. Sedangkan produk 2 mengalami penurunan *profit* dari Rp3.102.500,00 sampai Rp684.275.,00. Kemudian mengalami peningkatan sampai Rp17.347.570,00 dan mengalami penurunan kembali sampai Rp2.372.500,00. Sedangkan pada gambar 5.37 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.37 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp2.607.377.500,00 sampai Rp2.171.568.500,00. Parameter *c* yang digunakan pada produk 1 adalah 31 sampai 35 dan pada produk 2 adalah 11 sampai 15. Berdasarkan grafik pada gambar 5.37 dapat diketahui bahwa semakin besar nilai *purchasing cost* maka *total profit* akan semakin kecil.



Gambar 5. 36 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter *c*



Gambar 5. 37 Grafik *Total Profit* Skenario 2 Produk Substitusi Parameter *c*

5.3.5 Analisis Sensitivitas Parameter *h*

Salah satu parameter yang digunakan dalam analisis sensitivitas adalah *h* (holding cost). Analisis sensitivitas untuk parameter *h* dibagi ke dalam 2 skenario dengan 2 tipe produk. Analisis sensitivitas dilakukan dengan mengubah beberapa nilai pada parameter *h* terhadap produk yang diujikan. Parameter *h* diubah untuk

mengetahui pengaruh parameter tersebut terhadap *total profit* pada 2 skenario. Berikut akan dibahas analisis sensitivitas untuk parameter *h*.

5.3.5.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

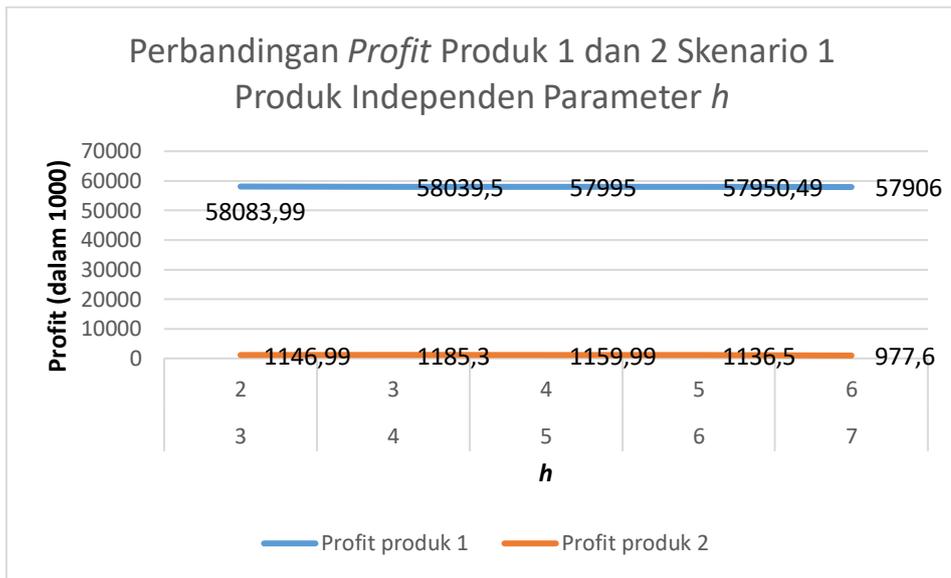
a. Produk Independen

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.26.

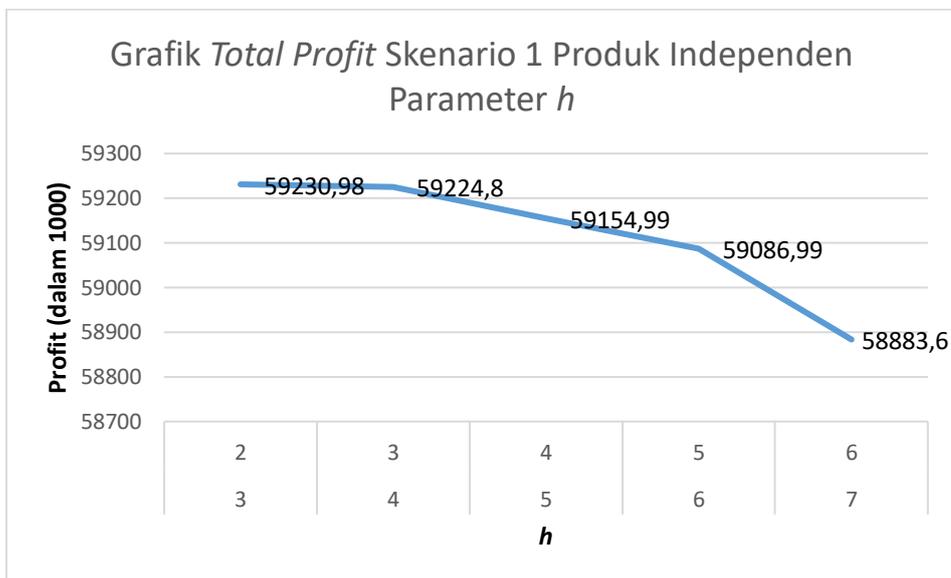
Tabel 5. 26 *Output* Hasil Analisis Sensitivitas Parameter *h*

<i>h</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
3	2	24	15	100	51	4	9	58.083,99	1.146,99	59.230,98
4	3	24	16	100	51	4	9	58.039,5	1.185,3	59.224,8
5	4	24	15	100	50	4	8	57.995	1.159,99	59.154,99
6	5	24	15	100	50	4	8	57.950,49	1.136,5	59.086,99
7	6	24	16	100	51	4	10	57.906	977,6	58.883,6

Grafik hasil analisis sensitivitas parameter *h* produk independen skenario 1 terlihat pada gambar 5.38. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.38. *Profit* produk 1 mengalami penurunan dari Rp58.083.990,00 sampai Rp57.906.000,00. Sedangkan produk 2 juga mengalami penurunan *profit* dari Rp6.938.750,00 sampai -Rp2.284.090,00. Sedangkan pada gambar 5.39 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.39 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp59.230.980,00 sampai Rp58.883.600,00. Parameter *h* yang digunakan pada produk 1 adalah Rp3.000,00 sampai Rp7.000,00 dan pada produk 2 adalah Rp2.000,00 sampai Rp6.000,00. Berdasarkan grafik pada gambar 5.39 dapat diketahui bahwa semakin besar nilai *holding cost* maka *total profit* akan semakin kecil.



Gambar 5. 38 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Independen Parameter h



Gambar 5. 39 Grafik *Total Profit* Skenario 1 Produk Independen Parameter h

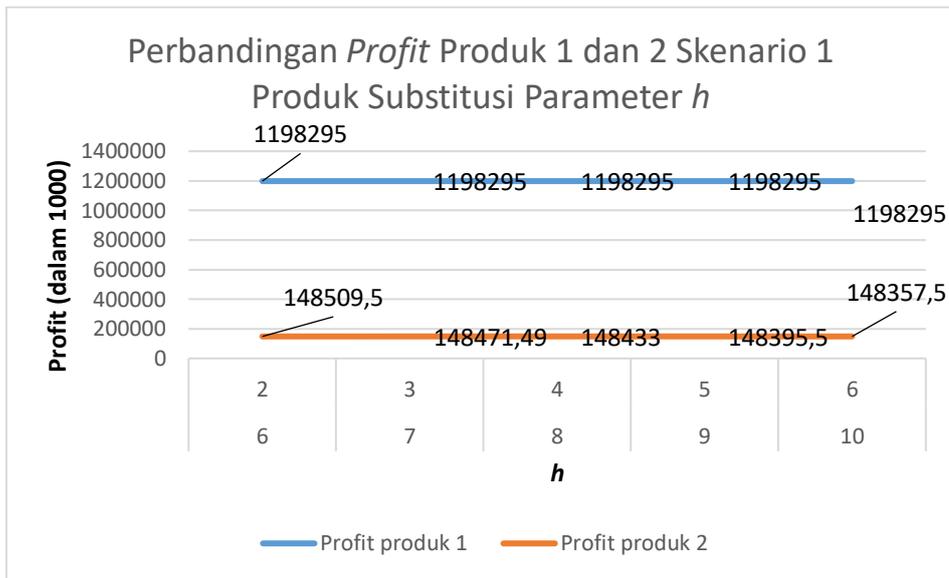
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.27.

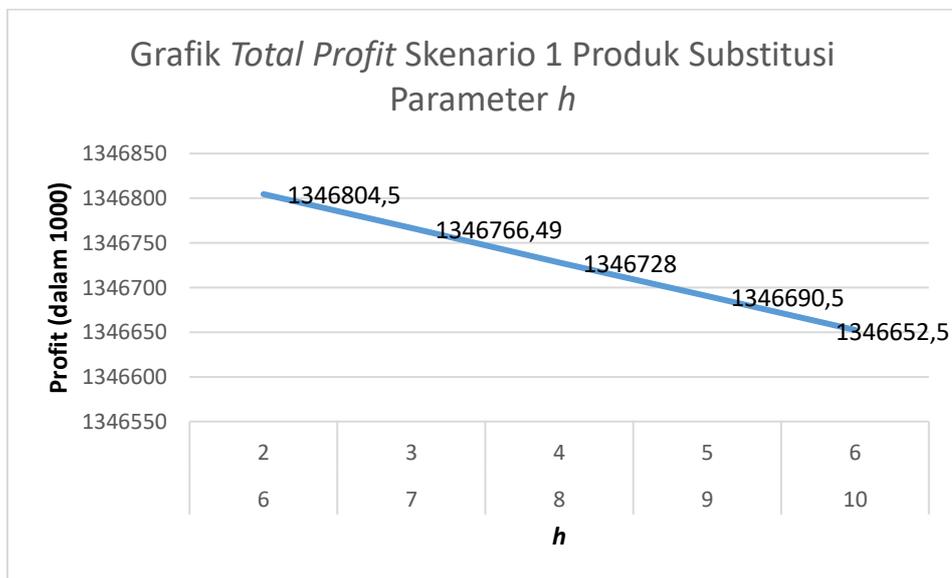
Tabel 5. 27 *Output* Hasil Analisis Sensitivitas Parameter *h*

<i>h</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
6	2	55	31	150	76	1	2	1.198.295	148.509,5	1.346.805
7	3	55	31	150	76	1	2	1.198.295	148.471,49	1.346.766
8	4	55	31	150	76	1	2	1.198.295	148.433	1.346.728
9	5	55	31	150	76	1	2	1.198.295	148.395,5	1.346.691
10	6	55	31	150	76	1	2	1.198.295	148.357,5	1.346.653

Grafik hasil analisis sensitivitas parameter *h* produk substitusi skenario 1 terlihat pada gambar 5.40. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.40. *Profit* produk 1 tidak mengalami penurunan maupun peningkata. Nilai *profit* datar di angka Rp1.198.295.000,00. Sedangkan produk 2 mengalami penurunan *profit* dari Rp148.609.500,00 sampai Rp148.357.500,00. Sedangkan pada gambar 5.41 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.41 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp1.346.804.500,00 sampai Rp1.346.652.500,00. Parameter *h* yang digunakan pada produk 1 adalah Rp6.000,00 sampai Rp10.000,00 dan pada produk 2 adalah Rp2.000,00 sampai Rp6.000,00. Berdasarkan grafik pada gambar 5.41 dapat diketahui bahwa semakin besar nilai *holding cost* maka *total profit* akan semakin kecil.



Gambar 5. 40 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter h



Gambar 5. 41 Grafik *Total Profit* Skenario 1 Produk Substitusi Parameter h

5.3.5.2 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

a. Produk Independen

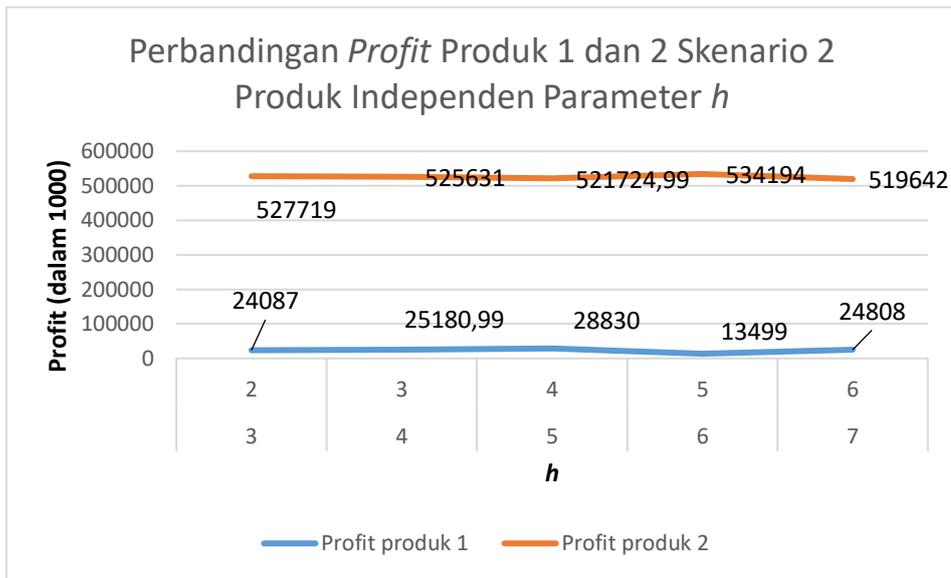
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan

dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.28.

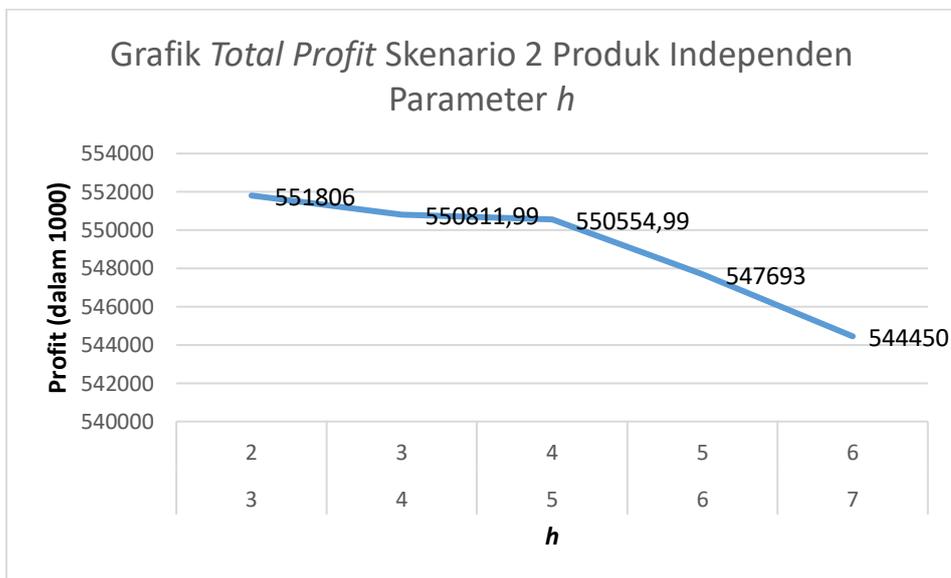
Tabel 5. 28 *Output* Hasil Analisis Sensitivitas Parameter *h*

<i>h</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
3	2	26	10	8	291	1	1	24.087	527.719	551.806
4	3	25	10	9	290	1	1	25.180,99	525.631	550.812
5	4	24	10	11	288	1	1	28.830	521.724,99	550.555
6	5	31	10	4	295	1	1	13.499	534.194	547.693
7	6	27	10	12	287	2	1	24.808	519.642	544.450

Grafik hasil analisis sensitivitas parameter *h* produk independen skenario 2 terlihat pada gambar 5.42. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.42. *Profit* produk 1 mengalami penurunan dari Rp527.719.000,00 sampai Rp521.724.990,00. Kemudian mengalami peningkatan sampai Rp534.194.000,00 dan mengalami penurunan Kembali sampai Rp519.642.000,00. Sedangkan produk 2 juga mengalami peningkatan dari Rp24.087.000,00 sampai Rp28.830.000,00. Kemudian mengalami penurunan sampai Rp13.499.000,00 dan mengalami peningkatan Kembali sampai Rp24.808.000,00. Sedangkan pada gambar 5.43 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.43 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp551.806.000,00 sampai Rp544.450.000,00. Parameter *h* yang digunakan pada produk 1 adalah Rp3.000,00 sampai Rp7.000,00 dan pada produk 2 adalah Rp2.000,00 sampai Rp6.000,00. Berdasarkan grafik pada gambar 5.43 dapat diketahui bahwa semakin besar nilai *holding cost* maka *total profit* akan semakin kecil.



Gambar 5. 42 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter h



Gambar 5. 43 Grafik *Total Profit* Skenario 2 Produk Independen Parameter h

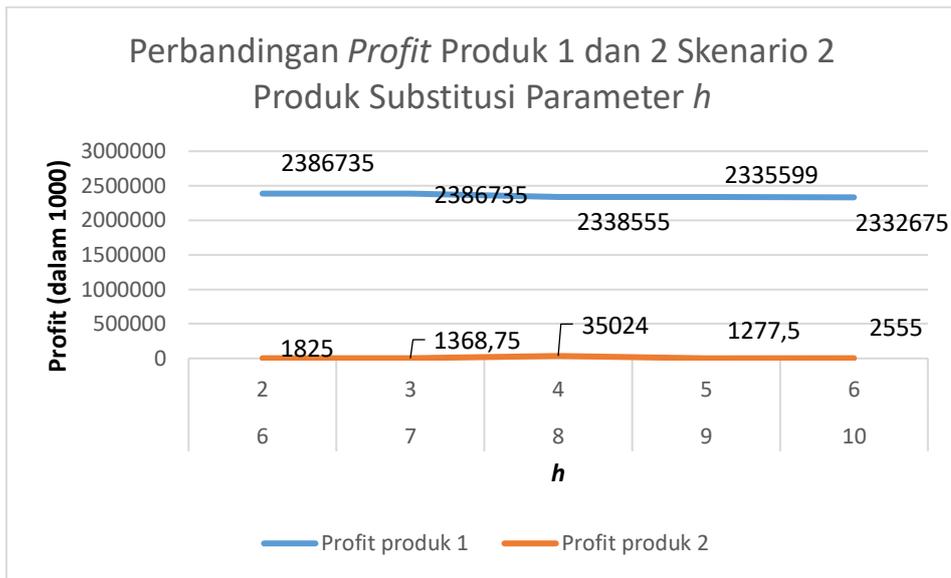
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.29.

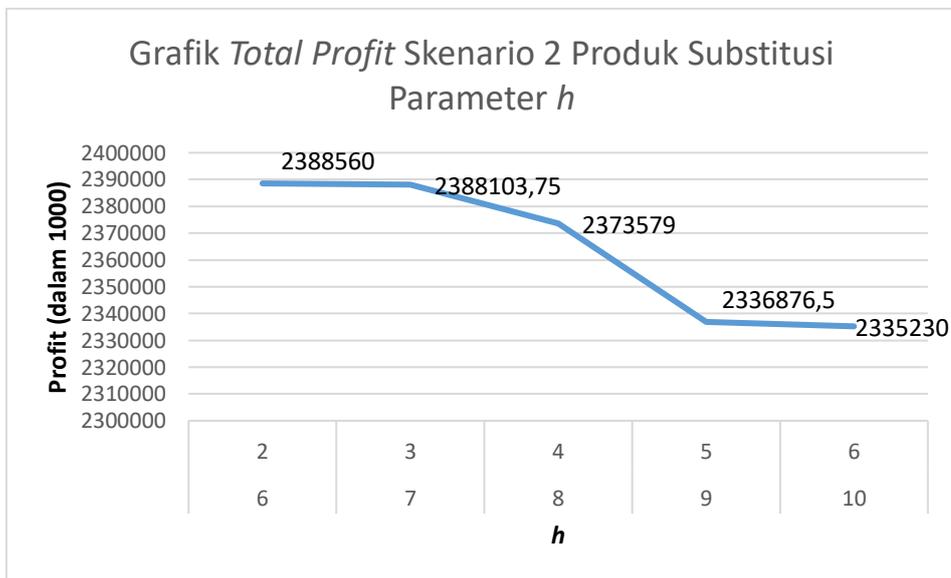
Tabel 5. 29 *Output* Hasil Analisis Sensitivitas Parameter *h*

<i>h</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
6	2	55	35	298	1	1	3	2.386.735	1.825	2.388.560
7	3	55	35	298	1	1	4	2.386.735	1.368,75	2.388.104
8	4	55	34	292	7	1	1	2.338.555	3.5024	2.373.579
9	5	55	35	298	1	1	4	2.335.599	1.277,5	2.336.877
10	6	55	34	298	1	1	2	2.332.675	2.555	2.335.230

Grafik hasil analisis sensitivitas parameter *h* produk substitusi skenario 2 terlihat pada gambar 5.44. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.44. *Profit* produk 1 mengalami penurunan dari Rp2.386.735.000,00 sampai Rp2.338.555.000,00. Kemudian mengalami peningkatan sampai Rp2.335.599.000,00 dan mengalami peningkatan Kembali sampai Rp2.332.675.000,00. Sedangkan produk 2 mengalami penurunan *profit* dari Rp1.825.000,00 sampai Rp1.368.750,00. Kemudian mengalami peningkatan sampai Rp35.024.000,00 dan mengalami penurunan sampai Rp1.277.500. Sampai akhirnya mengalami peningkatan Kembali sampai Rp2.555.000. Sedangkan pada gambar 5.45 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.45 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp2.388.560.000,00 sampai Rp2.335.230.000,00. Parameter *h* yang digunakan pada produk 1 adalah Rp6.000,00 sampai Rp10.000,00 dan pada produk 2 adalah Rp2.000,00 sampai Rp6.000,00. Berdasarkan grafik pada gambar 5.45 dapat diketahui bahwa semakin besar nilai *holding cost* maka *total profit* akan semakin kecil.



Gambar 5. 44 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter h



Gambar 5. 45 Grafik *Total Profit* Skenario 2 Produk Substitusi Parameter h

5.3.6 Analisis Sensitivitas Parameter o

Salah satu parameter yang digunakan dalam analisis sensitivitas adalah o (order cost). Analisis sensitivitas untuk parameter o dibagi ke dalam 2 skenario dengan 2 tipe produk. Analisis sensitivitas dilakukan dengan mengubah beberapa

nilai pada parameter o terhadap produk yang diujikan. Parameter o diubah untuk mengetahui pengaruh parameter tersebut terhadap *total profit* pada 2 skenario. Berikut akan dibahas analisis sensitivitas untuk parameter o .

5.3.6.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

a. Produk Independen

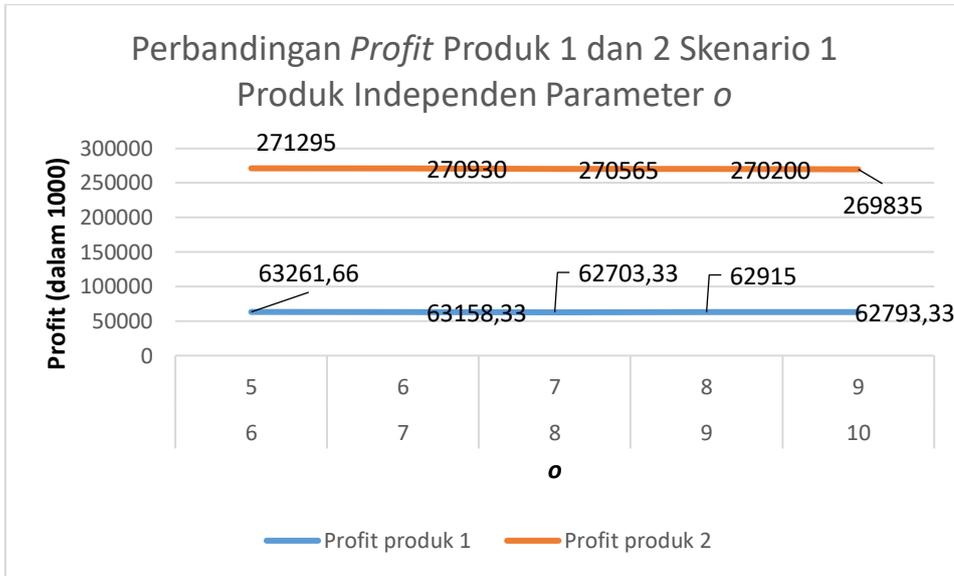
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.30.

Tabel 5. 30 *Output* Hasil Analisis Sensitivitas Parameter o

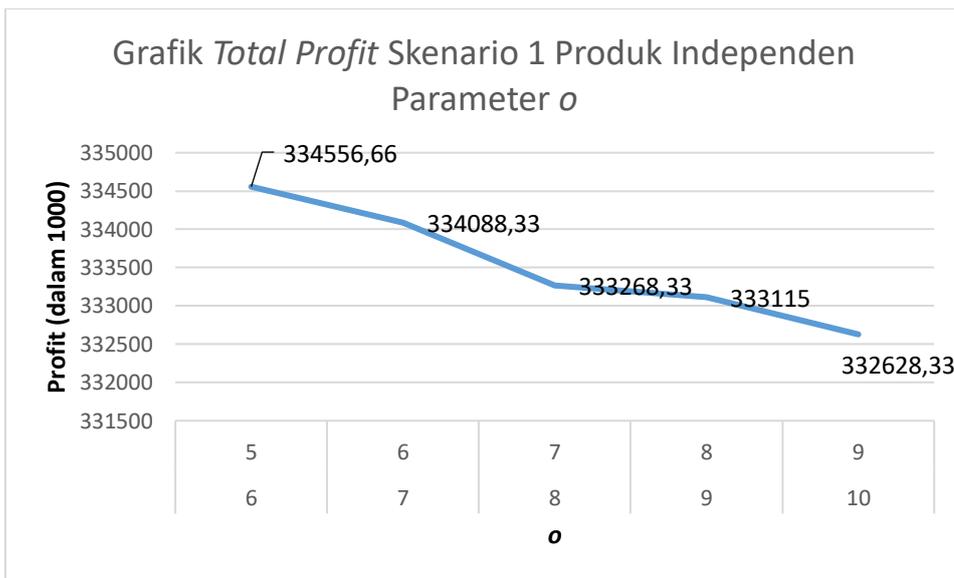
o										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
6	5	24	10	96	150	3	1	63.261,66	271.295	334.556,66
7	6	24	10	92	150	3	1	63.158,33	270.930	334.088,33
8	7	24	10	85	150	3	1	62.703,33	270.565	333.268,33
9	8	24	10	92	150	3	1	62.915	270.200	333.115
10	9	24	10	92	150	3	1	62.793,33	269.835	332.628,33

Grafik hasil analisis sensitivitas parameter o produk independen skenario 1 terlihat pada gambar 5.46. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.46. *Profit* produk 1 mengalami penurunan dari Rp63.261.660,00 sampai Rp62.703.330,00. Kemudian mengalami peningkatan sampai Rp62.915.000,00 dan mengalami penurunan kembali sampai Rp62.793.330,00. Sedangkan produk 2 juga mengalami penurunan *profit* dari Rp271.295.000,00 sampai Rp269.835.000,00. Sedangkan pada gambar 5.47 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.47 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp334.556.660,00 sampai Rp332.628.330,00. Parameter o yang digunakan pada produk 1 adalah Rp6.000,00 sampai Rp10.000,00 dan pada produk 2 adalah Rp5.000,00 sampai Rp9.000,00. Berdasarkan grafik pada gambar 5.47 dapat

diketahui bahwa semakin besar nilai *order cost* maka *total profit* akan semakin kecil.



Gambar 5. 46 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Independen Parameter o



Gambar 5. 47 Grafik *Total Profit* Skenario 1 Produk Independen Parameter o

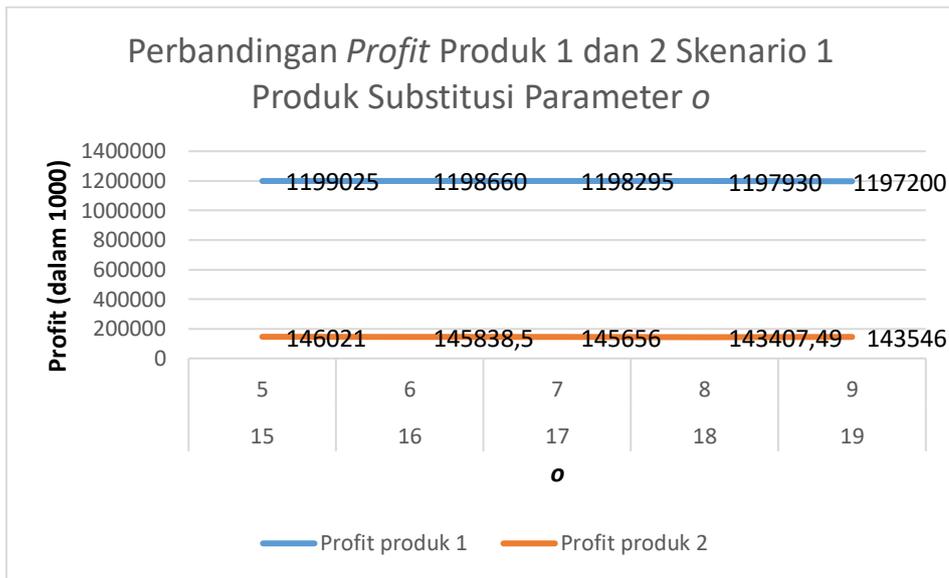
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.31.

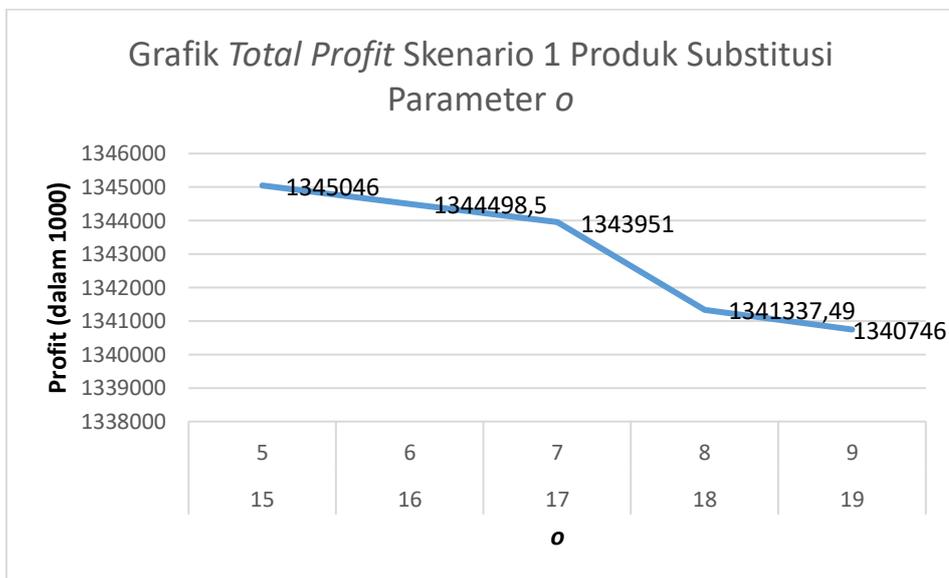
Tabel 5. 31 *Output* Hasil Analisis Sensitivitas Parameter *o*

<i>o</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
15	5	55	31	150	81	1	2	1.199.025	146.021	1.345.046
16	6	55	31	150	81	1	2	1.198.660	145.838,5	1.344.499
17	7	55	31	150	81	1	2	1.198.295	145.656	1.343.951
18	8	55	31	150	64	1	2	1.197.930	143.407,49	1.341.337
19	9	55	31	150	70	1	2	1.197.200	143.546	1.340.746

Grafik hasil analisis sensitivitas parameter *o* produk substitusi skenario 1 terlihat pada gambar 5.48. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.48. *Profit* produk 1 mengalami penurunan dari Rp1.199.025.000,00 sampai Rp1.197.200.000,00. Sedangkan produk 2 juga mengalami penurunan *profit* dari Rp146.021.000,00 sampai Rp143.546.000,00. Sedangkan pada gambar 5.49 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.49 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp1.345.046.000,00 sampai Rp1.340.746.000,00. Parameter *o* yang digunakan pada produk 1 adalah Rp15.000,00 sampai Rp19.000,00 dan pada produk 2 adalah Rp5.000,00 sampai Rp9.000,00. Berdasarkan grafik pada gambar 5.49 dapat diketahui bahwa semakin besar nilai *order cost* maka *total profit* akan semakin kecil.



Gambar 5. 48 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter *o*



Gambar 5. 49 Grafik *Total Profit* Skenario 1 Produk Substitusi Parameter *o*

5.3.6.2 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

a. Produk Independen

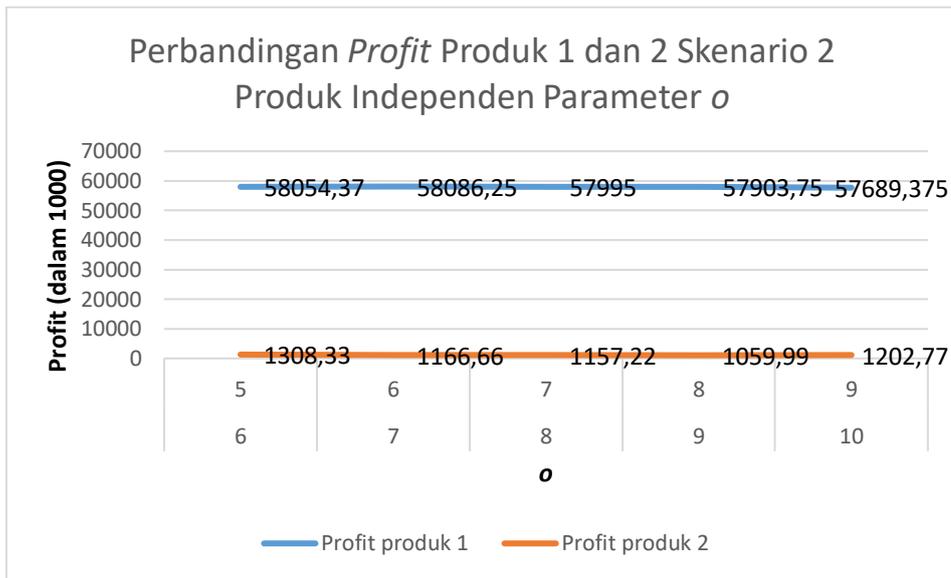
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan

dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.32.

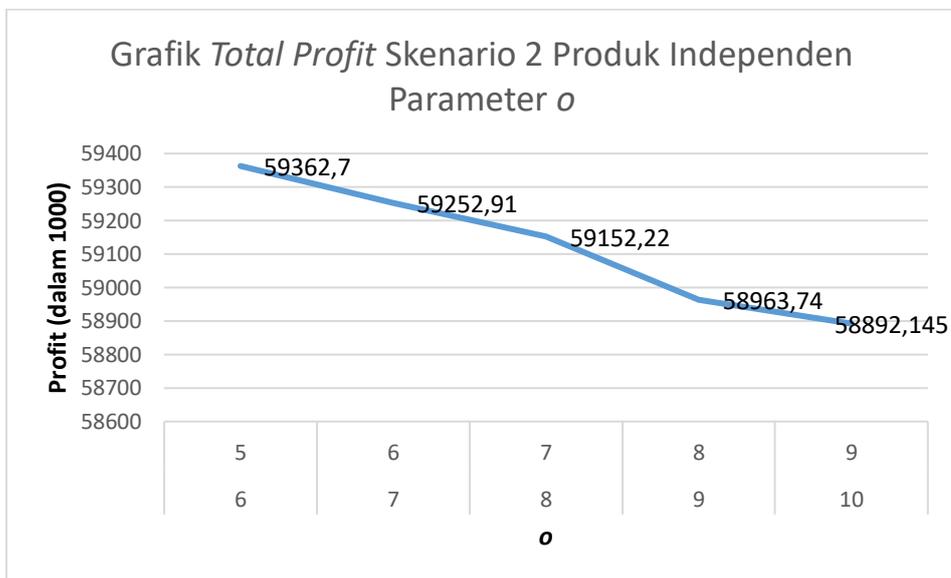
Tabel 5. 32 *Output* Hasil Analisis Sensitivitas Parameter *o*

<i>o</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
6	5	24	15	97	50	4	9	58.054,37	1.308,33	59.362,7
7	6	24	18	100	51	4	9	58.086,25	1.166,66	59.252,91
8	7	24	16	100	51	4	9	57.995	1.157,22	59.152,22
9	8	24	15	100	51	4	9	57.903,75	1.059,99	58.963,74
10	9	24	16	97	50	4	9	57.689,38	1.202,77	58.892,15

Grafik hasil analisis sensitivitas parameter *o* produk independen skenario 2 terlihat pada gambar 5.50. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.50. *Profit* produk 1 mengalami penurunan dari Rp58.043.370,00 sampai Rp57.689.375,00. Sedangkan produk 2 juga mengalami penurunan *profit* dari Rp1.308.330,00 sampai Rp1.059.990,00. Kemudian mengalami peningkatan sampai Rp1.202.770,00. Sedangkan pada gambar 5.51 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.51 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp59.362.700,00 sampai Rp58.892.145,00. Parameter *o* yang digunakan pada produk 1 adalah Rp6.000,00 sampai Rp10.000,00 dan pada produk 2 adalah Rp5.000,00 sampai Rp9.000,00. Berdasarkan grafik pada gambar 5.51 dapat diketahui bahwa semakin besar nilai *order cost* maka *total profit* akan semakin kecil.



Gambar 5. 50 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter *o*



Gambar 5. 51 Grafik *Total Profit* Skenario 2 Produk Independen Parameter *o*

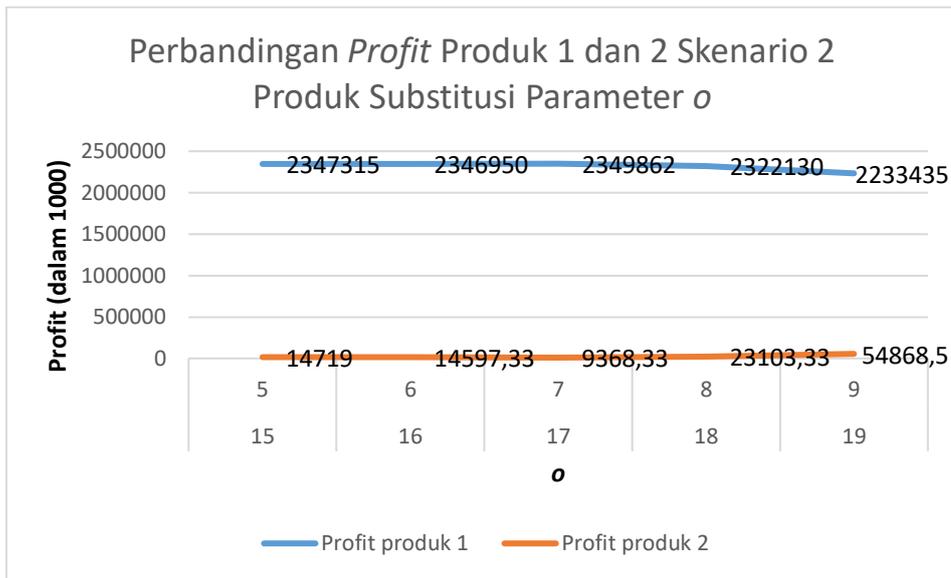
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.33.

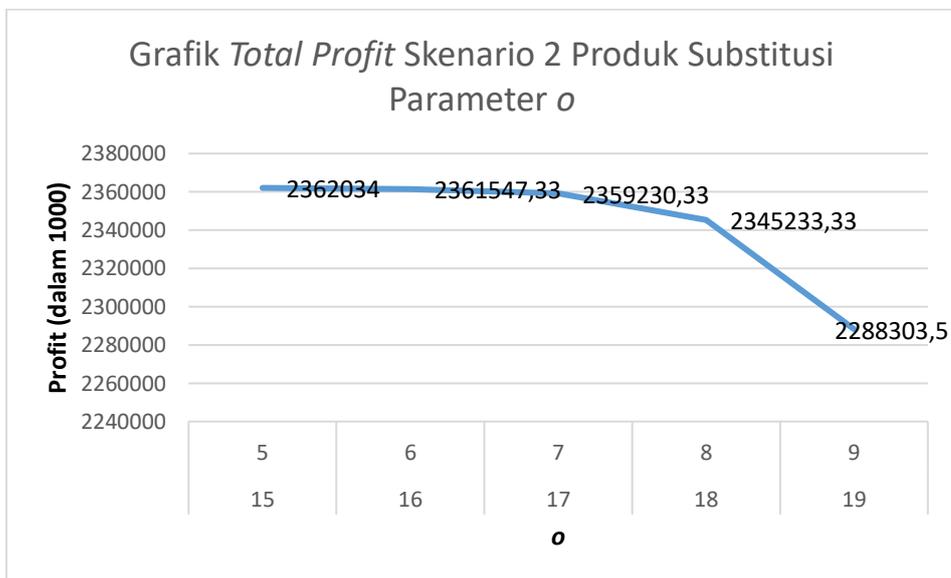
Tabel 5. 33 *Output* Hasil Analisis Sensitivitas Parameter *o*

<i>o</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
15	5	55	34	293	6	1	3	2.347.315	14.719	2.362.034
16	6	55	34	293	6	1	3	2.346.950	14.597,33	2.361.547
17	7	55	34	295	4	1	3	2.349.862	9.368,33	2.359.230
18	8	55	35	290	9	1	3	2.322.130	23.103,33	2.345.233
19	9	55	34	279	20	1	2	2.233.435	54.868,5	2.288.304

Grafik hasil analisis sensitivitas parameter *o* produk substitusi skenario 2 terlihat pada gambar 5.52. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.52. *Profit* produk 1 mengalami penurunan dari Rp2.347.315.000,00 sampai Rp2.233.435.000,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp14.719.000,00 sampai Rp14.597.330,00. Kemudian mengalami penurunan sampai Rp9.368.330,00 dan mengalami peningkatan kembali sampai Rp54.868.500,00 Sedangkan pada gambar 5.53 menggambarkan *total profit* kedua gabungan kedua produk. Berdasarkan grafik pada gambar 5.53 dapat dilihat bahwa *profit* keseluruhan mengalami penurunan dari Rp2.362.034.000,00 sampai Rp2.288.303.500,00. Parameter *o* yang digunakan pada produk 1 adalah Rp15.000,00 sampai Rp19.000,00 dan pada produk 2 adalah Rp5.000,00 sampai Rp9.000,00. Berdasarkan grafik pada gambar 5.53 dapat diketahui bahwa semakin besar nilai *order cost* maka *total profit* akan semakin kecil.



Gambar 5. 52 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter *o*



Gambar 5. 53 Grafik *Total Profit* Skenario 2 Produk Substitusi Parameter *o*

5.3.7 Analisis Sensitivitas Parameter *s*

Salah satu parameter yang digunakan dalam analisis sensitivitas adalah *s* (salvage price). Analisis sensitivitas untuk parameter *s* dibagi ke dalam 2 skenario dengan 2 tipe produk. Analisis sensitivitas dilakukan dengan mengubah beberapa nilai pada parameter *s* terhadap produk yang diujikan. Parameter *s* diubah untuk

mengetahui pengaruh parameter tersebut terhadap *total profit* pada 2 skenario. Berikut akan dibahas analisis sensitivitas untuk parameter *s*.

5.3.7.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

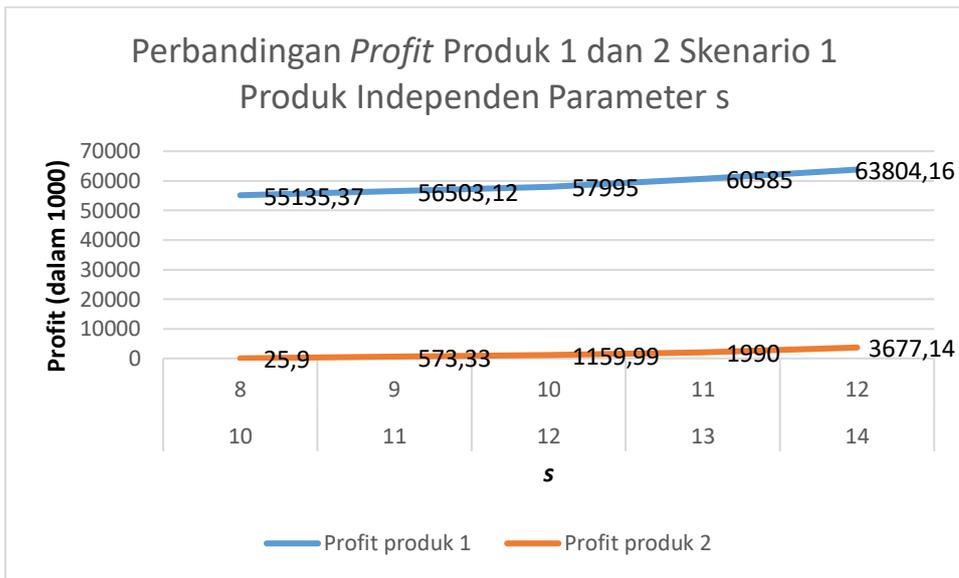
a. Produk Independen

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.34.

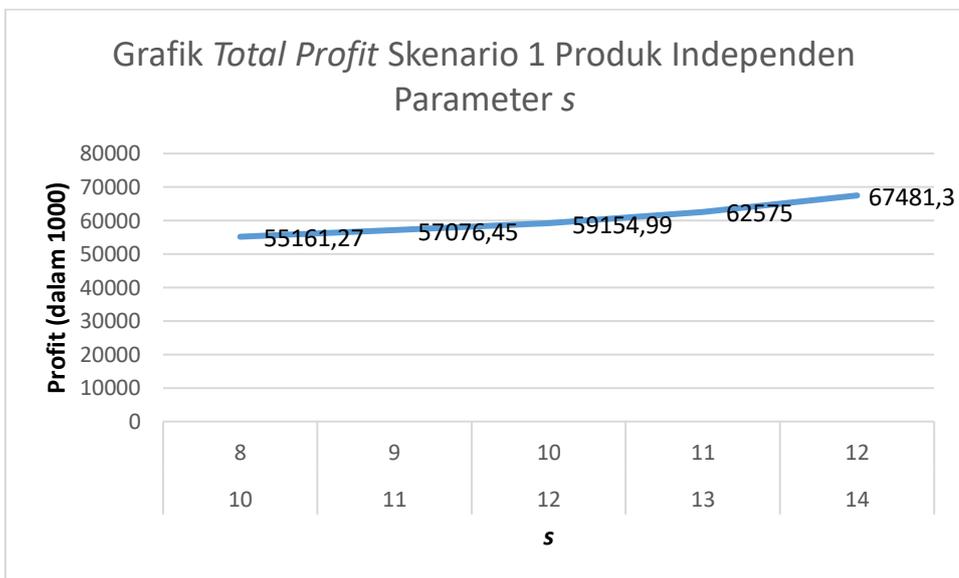
Tabel 5. 34 *Output* Hasil Analisis Sensitivitas Parameter *s*

<i>s</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
10	8	24	15	97	51	4	11	55.135,37	25,9	55.161,27
11	9	24	15	97	51	4	9	56.503,12	573,33	57.076,45
12	10	24	15	100	50	4	8	57.995	1.159,99	59.154,99
13	11	24	17	96	51	3	9	60.585	1.990	62.575
14	12	24	18	97	51	3	7	63.804,16	3.677,14	67.481,3

Grafik hasil analisis sensitivitas parameter *s* produk independen skenario 1 terlihat pada gambar 5.54. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.54. *Profit* produk 1 mengalami peningkatan dari Rp55.135.370,00 sampai Rp63.804.160,00. Sedangkan produk 2 juga mengalami peningkatan *profit* dari Rp25.900,00 sampai Rp3.677.140,00. Sedangkan pada gambar 5.55 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.55 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp55.161.270,00 sampai Rp67.681.300,00. Parameter *s* yang digunakan pada produk 1 adalah Rp10.000,00 sampai Rp14.000,00 dan pada produk 2 adalah Rp8.000,00 sampai Rp12.000,00. Berdasarkan grafik pada gambar 5.55 dapat diketahui bahwa semakin besar nilai *salvage price* maka *total profit* akan semakin besar.



Gambar 5. 54 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Independen Parameter *s*



Gambar 5. 55 Grafik *Total Profit* Skenario 1 Produk Independen Parameter *s*

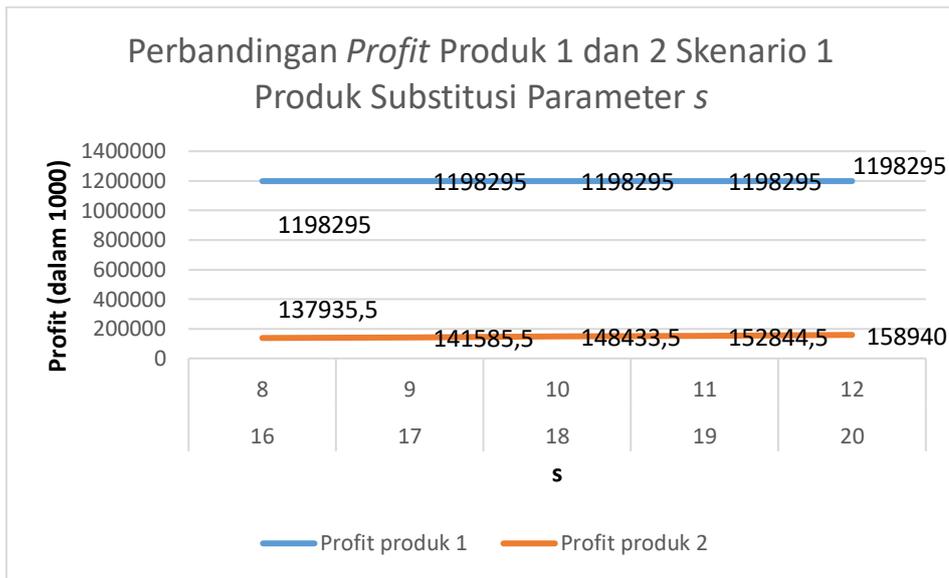
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.35.

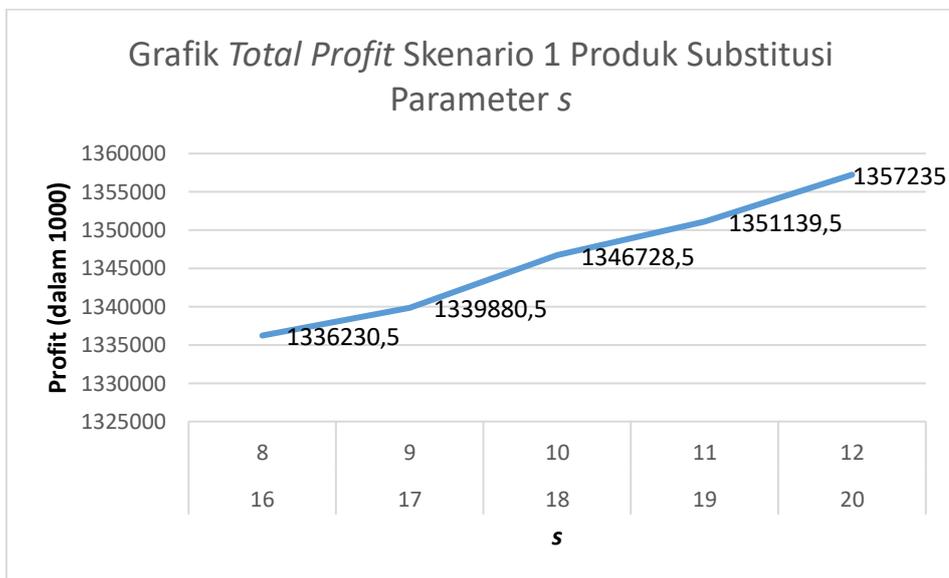
Tabel 5. 35 *Output* Hasil Analisis Sensitivitas Parameter *s*

<i>s</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
16	8	55	31	150	68	1	2	1.198.295	137.935,5	1.336.231
17	9	55	31	150	68	1	2	1.198.295	141.585,5	1.339.881
18	10	55	31	150	76	1	2	1.198.295	148.433,5	1.346.729
19	11	55	31	150	77	1	2	1.198.295	152.844,5	1.351.140
20	12	55	31	150	51	1	1	1.198.295	158.940	1.357.235

Grafik hasil analisis sensitivitas parameter *s* produk substitusi skenario 1 terlihat pada gambar 5.56. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.56. *Profit* produk 1 tidak mengalami peningkatan maupun penurunan, yaitu datar dinilai Rp1.198.295.000,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp137.935.500,00 sampai Rp158.940.000,00. Sedangkan pada gambar 5.57 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.57 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp1.336.230.500,00 sampai Rp1.357.235.000,00. Parameter *s* yang digunakan pada produk 1 adalah Rp16.000,00 sampai Rp20.000,00 dan pada produk 2 adalah Rp8.000,00 sampai Rp12.000,00. Berdasarkan grafik pada gambar 5.57 dapat diketahui bahwa semakin besar nilai *salvage price* maka *total profit* akan semakin besar.



Gambar 5. 56 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter *s*



Gambar 5. 57 Grafik *Total Profit* Skenario 1 Produk Substitusi Parameter *s*

5.3.7.2 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

a. Produk Independen

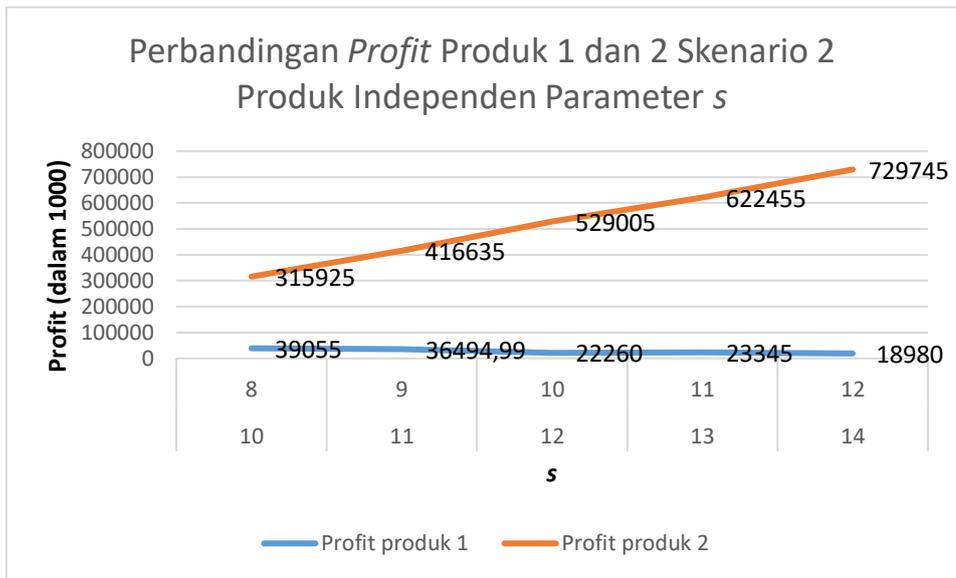
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan

dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.36.

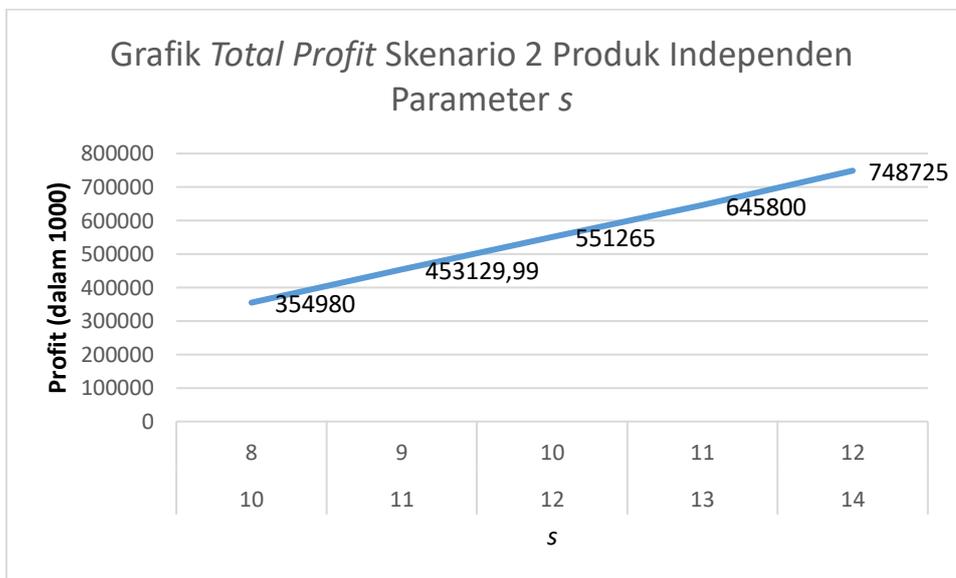
Tabel 5. 36 *Output* Hasil Analisis Sensitivitas Parameter *s*

<i>s</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
10	8	20	10	23	276	1	1	39.055	315.925	354.980
11	9	22	10	17	282	1	1	36.494,99	416.635	453.130
12	10	27	10	7	292	1	1	22.260	529.005	551.265
13	11	28	10	9	290	1	1	23.345	622.455	645.800
14	12	27	10	5	294	1	1	18.980	729.745	748.725

Grafik hasil analisis sensitivitas parameter *s* produk independen skenario 2 terlihat pada gambar 5.58. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.58. *Profit* produk 1 mengalami penurunan dari Rp39.055.000,00 sampai Rp22.260.000,00. Kemudian mengalami peningkatan samapai Rp 23.245.000,00 dan mengalami penurunan kembali sampai Rp18.980.000,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp315.925.000,00 sampai Rp729.745.000,00. Sedangkan pada gambar 5.59 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.59 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp354.980.000,00 sampai Rp748.725.000,00. Parameter *s* yang digunakan pada produk 1 adalah Rp10.000,00 sampai Rp14.000,00 dan pada produk 2 adalah Rp8.000,00 sampai Rp12.000,00. Berdasarkan grafik pada gambar 5.59 dapat diketahui bahwa semakin besar nilai *salvage price* maka *total profit* akan semakin besar.



Gambar 5. 58 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter *s*



Gambar 5. 59 Grafik *Total Profit* Skenario 2 Produk Independen Parameter *s*

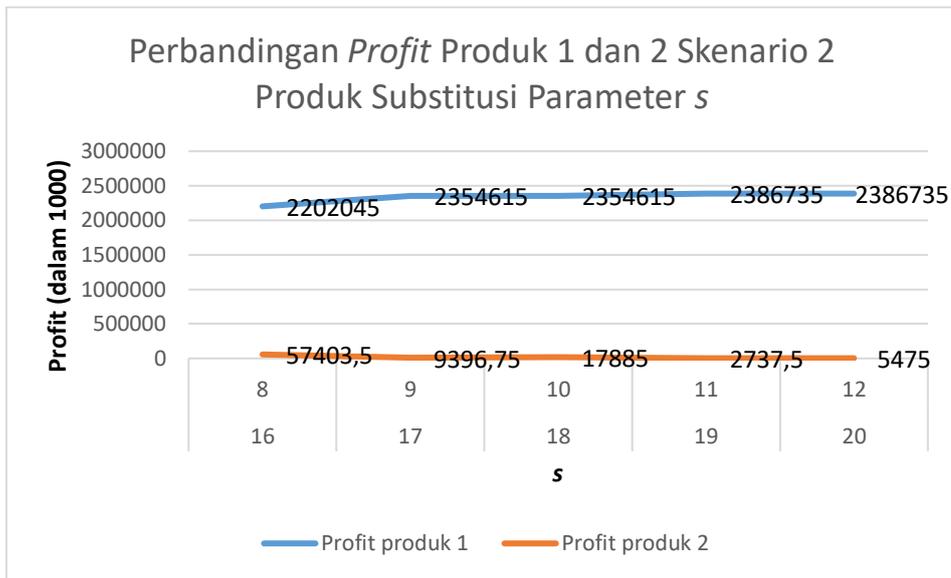
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala Rp1.000,00 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.37.

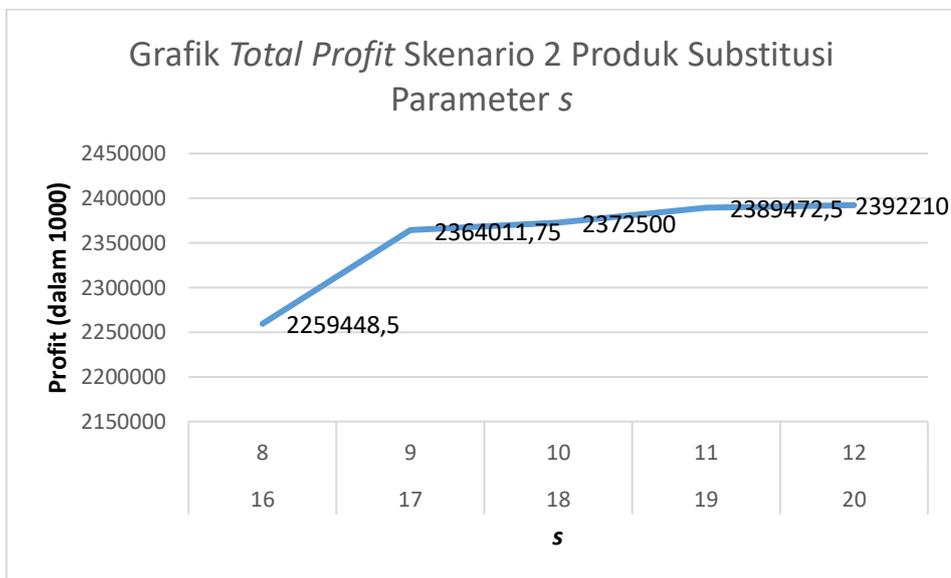
Tabel 5. 37 *Output Hasil Analisis Sensitivitas Parameter s*

s										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
16	8	55	34	275	24	1	2	2.202.045	57.403,5	2.259.449
17	9	55	35	294	5	1	4	2.354.615	9.396,75	2.364.012
18	10	55	34	294	5	1	2	2.354.615	17.885	2.372.500
19	11	55	35	298	1	1	2	2.386.735	2.737,5	2.389.473
20	12	55	35	298	1	1	1	2.386.735	5.475	2.392.210

Grafik hasil analisis sensitivitas parameter *s* produk substitusi skenario 2 terlihat pada gambar 5.60. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.60. *Profit* produk 1 mengalami peningkatan dari Rp2.202.045.000,00 sampai Rp2.386.735.000,00. Sedangkan produk 2 mengalami penurunan *profit* dari Rp57.403.500,00 sampai Rp2.737.500,00. Kemudian mengalami peningkatan sampai Rp5.475.000,00. Sedangkan pada gambar 5.61 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.61 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp2.259.448.500,00 sampai Rp2.392.210.000,00. Parameter *s* yang digunakan pada produk 1 adalah Rp16.000,00 sampai Rp20.000,00 dan pada produk 2 adalah Rp8.000,00 sampai Rp12.000,00. Berdasarkan grafik pada gambar 5.61 dapat diketahui bahwa semakin besar nilai *salvage price* maka *total profit* akan semakin besar.



Gambar 5. 60 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter *s*



Gambar 5. 61 Grafik *Total Profit* Skenario 2 Produk Substitusi Parameter *s*

5.3.8 Analisis Sensitivitas Parameter *w*

Salah satu parameter yang digunakan dalam analisis sensitivitas adalah *w* (*size of shelf space*). Analisis sensitivitas untuk parameter *w* dibagi ke dalam 2 skenario dengan 2 tipe produk. Analisis sensitivitas dilakukan dengan mengubah beberapa nilai pada parameter *w* terhadap produk yang diujikan. Parameter *w*

diubah untuk mengetahui pengaruh parameter tersebut terhadap *total profit* pada 2 skenario. Berikut akan dibahas analisis sensitivitas untuk parameter *w*.

5.3.8.1 Skenario 1: Batasan kapasitas gudang untuk masing-masing produk

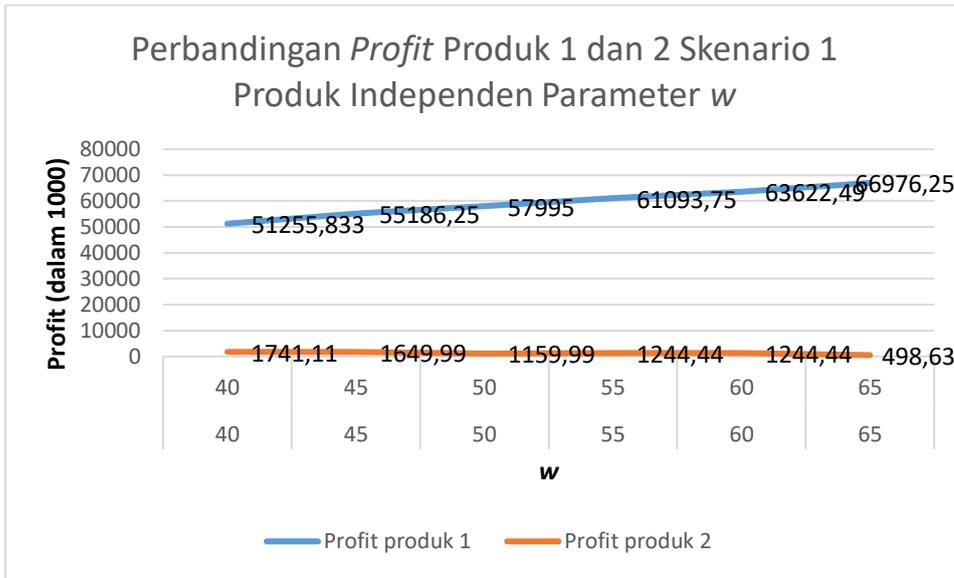
a. Produk Independen

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 5 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.38.

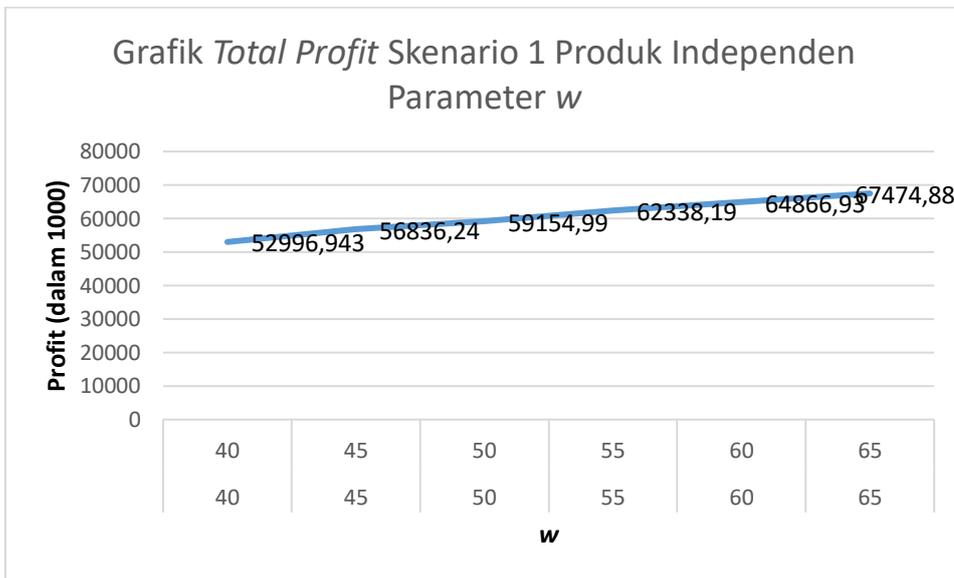
Tabel 5. 38 *Output* Hasil Analisis Sensitivitas Parameter *w*

<i>w</i>										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
40	40	24	17	69	43	3	9	51.255,83	1.741,11	52.996,943
45	45	25	17	88	46	4	9	55.186,25	1.649,99	56.836,24
50	50	24	15	100	50	4	8	57.995	1.159,99	59.154,99
55	55	24	16	104	62	4	9	61.093,75	1.244,44	62.338,19
60	60	25	16	106	62	4	9	63.622,49	1.244,44	64.866,93
65	65	25	17	114	72	4	11	66.976,25	498,63	67.474,88

Grafik hasil analisis sensitivitas parameter *w* produk independen skenario 1 terlihat pada gambar 5.62. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.62. *Profit* produk 1 mengalami peningkatan dari Rp51.255.833,00 sampai Rp66.976.250,00. Sedangkan produk 2 mengalami penurunan *profit* dari Rp1.741.110,00 sampai Rp498.630,00. Sedangkan pada gambar 5.63 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.63 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp52.996.943,00 sampai Rp67.474.880,00. Parameter *w* yang digunakan pada produk 1 dan 2 adalah 40 sampai 65. Berdasarkan grafik pada gambar 5.63 dapat diketahui bahwa semakin besar nilai *size of shelf space* maka *total profit* akan semakin besar.



Gambar 5. 62 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Independen Parameter w



Gambar 5. 63 Grafik *Total Profit* Skenario 1 Produk Independen Parameter w

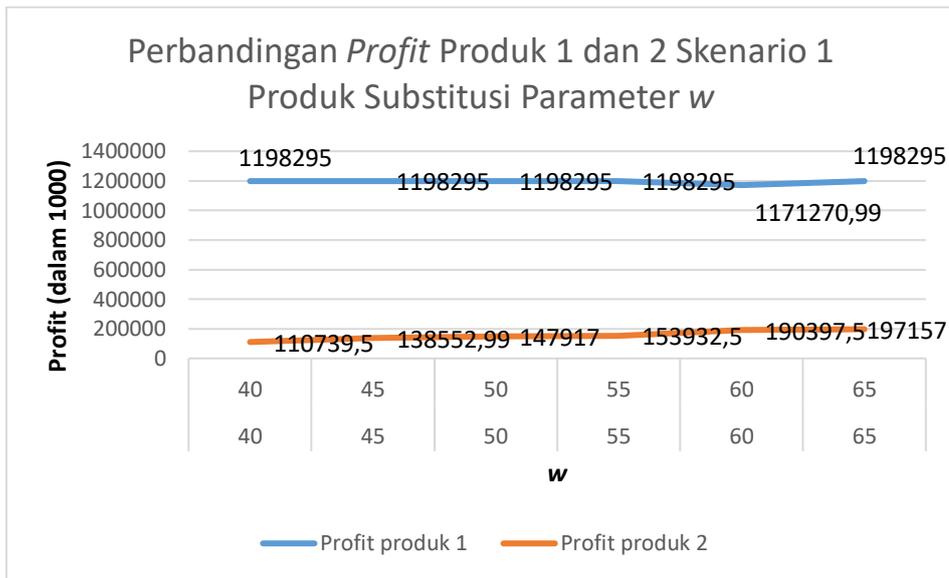
b. Produk Substitusi

Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan dengan skala 5 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.39.

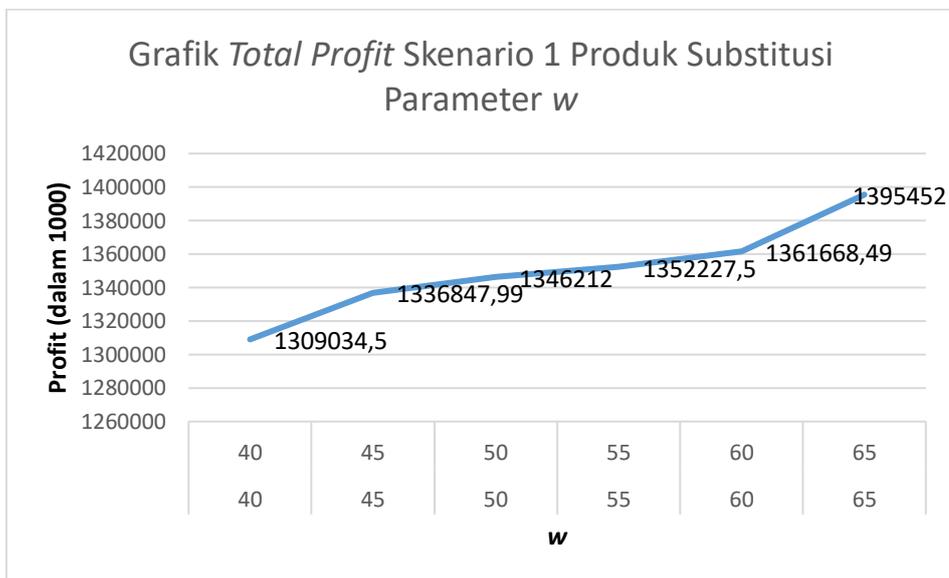
Tabel 5. 39 *Output* Hasil Analisis Sensitivitas Parameter w

w										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
40	40	55	32	150	59	1	2	1.198.295	110.739,5	1.309.035
45	45	55	31	150	46	1	1	1.198.295	138.552,99	1.336.848
50	50	55	31	150	77	1	2	1.198.295	147.917	1.346.212
55	55	55	31	150	80	1	2	1.198.295	153.932,5	1.352.228
60	60	55	30	150	96	2	3	1.171.271	190.397,5	1.361.668
65	65	55	30	150	97	1	2	1.198.295	197.157	1.395.452

Grafik Hasil analisis sensitivitas parameter w produk substitusi skenario 1 terlihat pada gambar 5.64. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.64. *Profit* produk 1 mengalami penurunan dari Rp1.198.295.000,00 sampai Rp1.171.270.990,00. Kemudian mengalami peningkatan sampai Rp1.198.295.000,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp110.739.500,00 sampai Rp117.157.000,00. Sedangkan pada gambar 5.65 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.65 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp1.309.034.500,00 sampai Rp1.395.452.000,00. Parameter w yang digunakan pada produk 1 dan 2 adalah 40 sampai 65. Berdasarkan grafik pada gambar 5.65 dapat diketahui bahwa semakin besar nilai *size of shelf space* maka *total profit* akan semakin besar.



Gambar 5. 64 Perbandingan *Profit* Produk 1 dan 2 Skenario 1 Produk Substitusi Parameter w



Gambar 5. 65 Grafik *Total Profit* Skenario 1 Produk Substitusi Parameter w

5.3.8.2 Skenario 2: Batasan kapasitas gudang untuk seluruh produk (*aggregate*)

a. Produk Independen

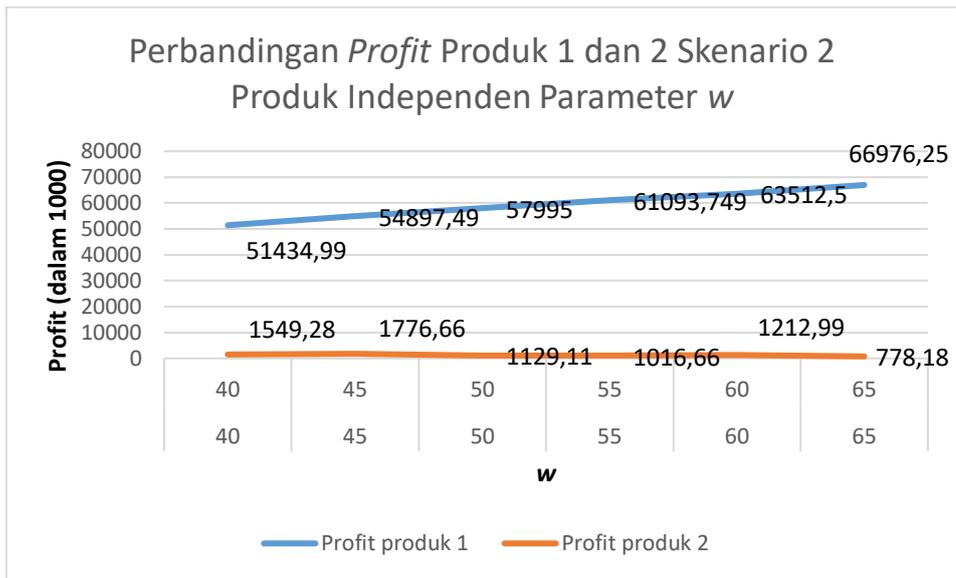
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan

dengan skala 5 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.40.

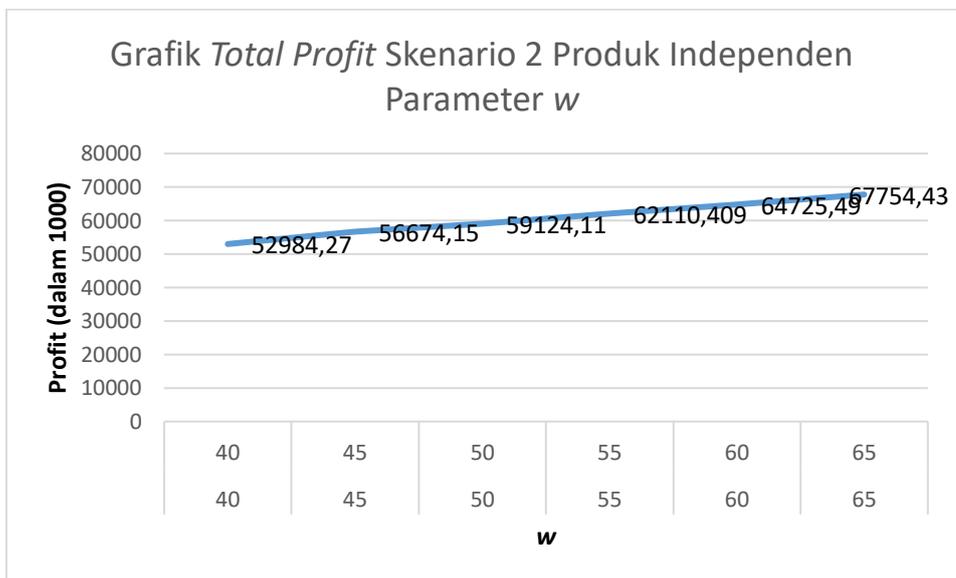
Tabel 5. 40 *Output* Hasil Analisis Sensitivitas Parameter w

w										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
40	40	25	16	66	43	3	7	51.434,99	1.549,28	52.984,27
45	45	24	17	96	45	4	9	54.897,49	1.776,66	56.674,15
50	50	24	18	100	51	4	9	57.995	1.129,11	59.124,11
55	55	24	16	104	56	4	9	61.093,75	1.016,66	62.110,41
60	60	25	16	110	63	4	10	63.512,5	1.212,99	64.725,49
65	65	25	16	114	66	4	11	66.976,25	778,18	67.754,43

Grafik hasil analisis sensitivitas parameter w produk independen skenario 2 terlihat pada gambar 5.66 Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.66 *Profit* produk 1 mengalami peningkatan dari Rp51.434.990,00 sampai Rp66.976.250,00. Sedangkan produk 2 mengalami peningkatan *profit* dari Rp1.549.280,00 sampai Rp1.776.660,00. Kemudian mengalami penurunan sampai Rp1.016.660,00 dan mengalami peningkatan kembali sampai Rp1.212.990,00. setelah itu, mengalami penurunan kembali sampai Rp778.180,00. Sedangkan pada gambar 5.67 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.67 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp52.984.270,00 sampai Rp67.754.430,00. Parameter w yang digunakan pada produk 1 dan 2 adalah 40 sampai 65. Berdasarkan grafik pada gambar 5.67 dapat diketahui bahwa semakin besar nilai *size of shelf space* maka *total profit* akan semakin besar.



Gambar 5. 66 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter w



Gambar 5. 67 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Independen Parameter w

b. Produk Substitusi

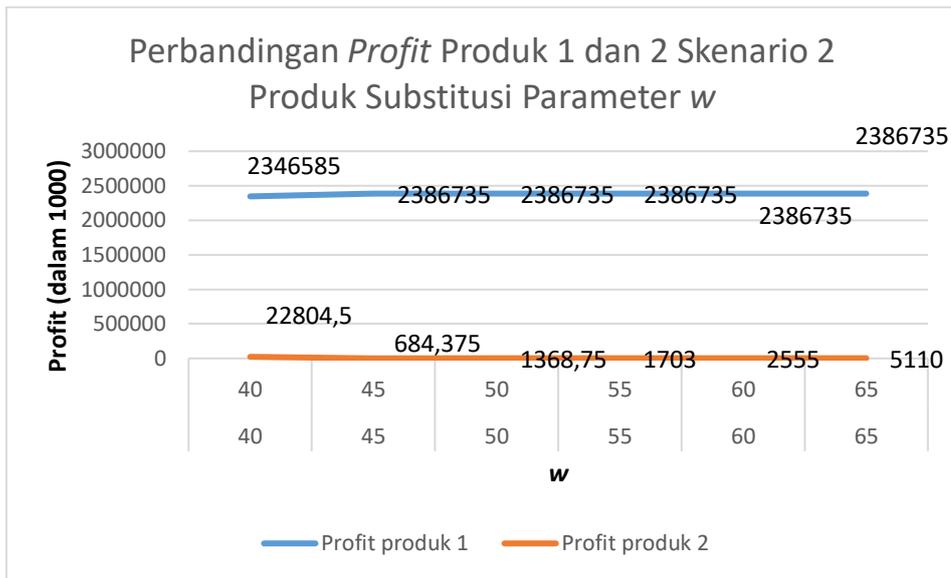
Analisis sensitivitas ini memiliki *output* akhir berupa *total profit* yang dihasilkan dari variabel. Beberapa perubahan terhadap parameter dilakukan

dengan skala 5 untuk setiap perubahan. Hasil dari analisis sensitivitas dapat dilihat pada tabel 5.41.

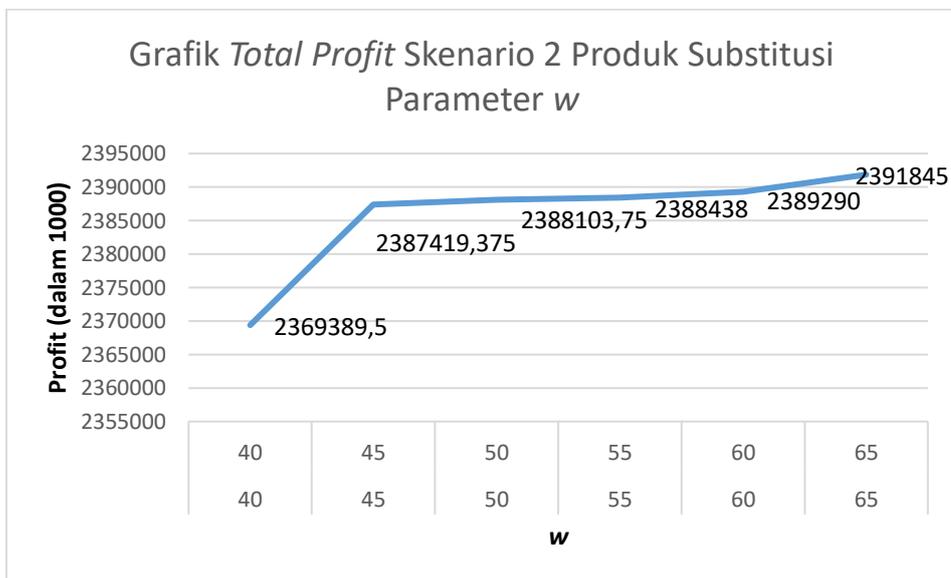
Tabel 5. 41 *Output* Hasil Analisis Sensitivitas Parameter w

w										
Parameter (Rp1.000)		P (Rp1.000)		Q		T		Profit (Rp1.000)		Total Profit (Rp1.000)
produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	produk 1	produk 2	
40	40	55	35	293	6	1	2	2.346.585	22.804,5	2.369.390
45	45	55	35	298	1	1	8	2.386.735	684,375	2.387.419
50	50	55	35	298	1	1	4	2.386.735	1.368,75	2.388.104
55	55	55	34	298	1	1	3	2.386.735	1.703	2.388.438
60	60	55	34	298	1	1	2	2.386.735	2.555	2.389.290
65	65	55	34	298	1	1	1	2.386.735	5.110	2.391.845

Grafik hasil analisis sensitivitas parameter w produk substitusi skenario 2 terlihat pada gambar 5.68. Gambar tersebut menggambarkan perbandingan *profit* produk 1 dan 2. Berdasarkan grafik pada gambar 5.68. *Profit* produk 1 mengalami peningkatan dari Rp2.346.585.000,00 sampai Rp2.386.735.000,00. Sedangkan produk 2 mengalami penurunan *profit* dari Rp22.804.500,00 sampai Rp648.375,00. Kemudian mengalami peningkatan kembali sampai Rp5.110.000,00. Sedangkan pada gambar 5.69 menggambarkan *total profit* gabungan kedua produk. Berdasarkan grafik pada gambar 5.69 dapat dilihat bahwa *profit* keseluruhan mengalami peningkatan dari Rp2.369.389.500,00 sampai Rp2.391.845.000,00. Parameter w yang digunakan pada produk 1 dan 2 adalah 40 sampai 65. Berdasarkan grafik pada gambar 5.69 dapat diketahui bahwa semakin besar nilai *size of shelf space* maka *total profit* akan semakin besar.



Gambar 5. 68 Perbandingan *Profit* Produk 1 dan 2 Skenario 2 Produk Substitusi Parameter w



Gambar 5. 69 Grafik *Total Profit* Skenario 2 Produk Substitusi Parameter w

5.4 Pembahasan

Pada sub bab sebelumnya telah dijelaskan mengenai hasil percobaan numerik terhadap 2 skenario (skenario 1 adalah batasan kapasitas gudang untuk masing-masing produk dan skenario 2 adalah batasan kapasitas gudang untuk seluruh produk/aggregate) dan 2 tipe produk (produk independent dan produk

substitusi). Berdasarkan hasil dari percobaan numerik tersebut dapat diketahui bahwa skenario 2 memiliki nilai *total profit* yang lebih besar untuk produk independen maupun substitusi. Untuk produk independen *total profit* skenario 2 adalah Rp59.069.280,00 lebih besar jika dibandingkan dengan skenario 1, yaitu Rp59.067.000,00. Selain itu, *total profit* skenario 2 pada produk substitusi juga memiliki nilai yang lebih besar, yaitu Rp2.334.390.290,00 dibandingkan dengan nilai *total profit* skenario 1, yaitu Rp1.341.620.670,00. Hal ini disebabkan karena skenario 2 dapat memaksimalkan seluruh kapasitas gudang untuk produk yang lebih menguntungkan.

Pada sub bab sebelumnya juga telah dijelaskan hasil analisis sensitivitas setiap parameter terhadap 2 skenario (skenario 1 adalah batasan kapasitas gudang untuk masing-masing produk dan skenario 2 adalah batasan kapasitas gudang untuk seluruh produk/*aggregate*) dan 2 tipe produk (produk independent dan produk substitusi). Parameter yang digunakan dalam analisis sensitivitas, yaitu: parameter α (*potential customer*), β (*shelf space efficiency of demand*), λ (*price efficiency of demand*), c (*purchasing cost*), h (*holding cost*), o (*ordering cost*), s (*salvage price*), dan w (*size of shelft space*). Berdasarkan hasil analisis sensitivitas tersebut, diketahui hal-hal sebagai berikut:

1. Parameter α (*potential customer*). Parameter α yang digunakan pada produk independen, yaitu: pada produk 1 dan 2 adalah 9000 sampai 13000. Sedangkan parameter α yang digunakan pada produk substitusi, yaitu: produk 1 dan 2 adalah 2000 sampai 6000. Pada skenario 1 produk independen, semakin besar nilai α yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter α adalah meningkatkan *total profit* gabungan. Untuk skenario 1 produk substitusi, semakin besar nilai α yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter α adalah meningkatkan *total profit* gabungan. Sedangkan pada skenario 2 produk independen, semakin besar nilai α yang digunakan juga menyebabkan semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter α adalah meningkatkan *total profit* gabungan. Untuk skenario 2 produk substitusi juga memiliki hasil yang sama, yaitu semakin besar nilai α yang digunakan

maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter α adalah meningkatkan *total profit* gabungan.

2. Parameter β (*shelf space efficiency of demand*). Parameter β yang digunakan pada produk independen dan substitusi, yaitu: produk 1 adalah 0,3 sampai 0,7 dan pada produk 2 adalah 0,2 sampai 0,6. Pada skenario 1 produk independen, semakin besar nilai β yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter β adalah meningkatkan *total profit* gabungan. Untuk skenario 1 produk substitusi, semakin besar nilai β yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter β adalah meningkatkan *total profit* gabungan. Sedangkan pada skenario 2 produk independen, semakin besar nilai β yang digunakan juga menyebabkan semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter β adalah meningkatkan *total profit* gabungan. Untuk skenario 2 produk substitusi juga memiliki hasil yang sama, yaitu semakin besar nilai β yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter β adalah meningkatkan *total profit* gabungan.
3. Parameter λ (*price efficiency of demand*). Parameter λ yang digunakan pada produk independen dan substitusi, yaitu: produk 1 adalah 0,08 sampai 0,13 dan pada produk 2 adalah 0,18 sampai 0,23. Pada skenario 1 produk independen, semakin besar nilai λ yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter λ adalah menurunkan *total profit* gabungan. Untuk skenario 1 produk substitusi, semakin besar nilai λ yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter λ adalah meningkatkan *total profit* gabungan. Sedangkan pada skenario 2 produk independen, semakin besar nilai λ yang digunakan menyebabkan semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter λ adalah menurunkan *total profit* gabungan. Untuk skenario 2 produk substitusi juga memiliki hasil yang sama, yaitu semakin besar nilai λ yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter λ adalah menurunkan *total profit* gabungan.

4. Parameter c (*purchasing cost*). Parameter c yang digunakan pada produk independen, yaitu: produk 1 adalah Rp13.500,00 sampai Rp17.500,00 dan pada produk 2 adalah Rp11.000,00 sampai Rp15.000,00. Sedangkan parameter c yang digunakan pada produk substitusi, yaitu: produk 1 adalah Rp31.000,00 sampai Rp35.000,00 dan pada produk 2 adalah Rp11.000,00 sampai Rp15.000,00. Pada skenario 1 produk independen, semakin besar nilai c yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter c adalah menurunkan *total profit* gabungan. Untuk skenario 1 produk substitusi, semakin besar nilai c yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter c adalah menurunkan *total profit* gabungan. Sedangkan pada skenario 2 produk independen, semakin besar nilai c yang digunakan menyebabkan semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter c adalah menurunkan *total profit* gabungan. Untuk skenario 2 produk substitusi juga memiliki hasil yang sama, yaitu semakin besar nilai c yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter c adalah menurunkan *total profit* gabungan.
5. Parameter h (*holding cost*). Parameter h yang digunakan pada produk independen, yaitu: produk 1 adalah Rp3.000,00 sampai Rp7.000,00 dan pada produk 2 adalah Rp2.000,00 sampai Rp6.000,00. Sedangkan parameter h yang digunakan pada produk substitusi, yaitu: produk 1 adalah Rp6.000,00 sampai Rp10.000,00 dan pada produk 2 adalah Rp2.000,00 sampai Rp6.000,00. Pada skenario 1 produk independen, semakin besar nilai h yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter h adalah menurunkan *total profit* gabungan. Untuk skenario 1 produk substitusi, semakin besar nilai h yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter h adalah menurunkan *total profit* gabungan. Sedangkan pada skenario 2 produk independen, semakin besar nilai h yang digunakan menyebabkan semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter h adalah menurunkan *total profit*

gabungan. Untuk skenario 2 produk substitusi juga memiliki hasil yang sama, yaitu semakin besar nilai h yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter h adalah menurunkan *total profit* gabungan.

6. Parameter o (*order cost*). Parameter o yang digunakan pada produk independen, yaitu: produk 1 adalah Rp6.000,00 sampai Rp10.000,00 dan pada produk 2 adalah Rp5.000,00 sampai Rp9.000,00. Sedangkan parameter o yang digunakan pada produk substitusi, yaitu: produk 1 adalah Rp15.000,00 sampai Rp19.000,00 dan pada produk 2 adalah Rp5.000,00 sampai Rp9.000,00. Pada skenario 1 produk independen, semakin besar nilai h yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter o adalah menurunkan *total profit* gabungan. Untuk skenario 1 produk substitusi, semakin besar nilai o yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter o adalah menurunkan *total profit* gabungan. Sedangkan pada skenario 2 produk independen, semakin besar nilai o yang digunakan menyebabkan semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter o adalah menurunkan *total profit* gabungan. Untuk skenario 2 produk substitusi juga memiliki hasil yang sama, yaitu semakin besar nilai o yang digunakan maka semakin kecil *total profit* gabungan yang dihasilkan. Sehingga, tren parameter o adalah menurunkan *total profit* gabungan.
7. Parameter s (*salvage price*). Parameter s yang digunakan pada produk independen, yaitu: produk 1 adalah Rp16.000,00 sampai Rp20.000,00 dan pada produk 2 adalah Rp8.000,00 sampai Rp12.000,00. Sedangkan parameter s yang digunakan pada produk substitusi, yaitu: produk 1 adalah Rp16.000,00 sampai Rp20.000,00 dan pada produk 2 adalah Rp8.000,00 sampai Rp12.000,00. Pada skenario 1 produk independen, semakin besar nilai s yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter s adalah meningkatkan *total profit* gabungan. Untuk skenario 1 produk substitusi, semakin besar nilai s yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan.

Sehingga, tren parameter s adalah meningkatkan *total profit* gabungan. Sedangkan pada skenario 2 produk independen, semakin besar nilai s yang digunakan menyebabkan semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter s adalah meningkatkan *total profit* gabungan. Untuk skenario 2 produk substitusi juga memiliki hasil yang sama, yaitu semakin besar nilai s yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter s adalah meningkatkan *total profit* gabungan.

8. Parameter w (*size of shelft space*). Parameter w yang digunakan pada produk independent dan substitusi, yaitu: produk 1 dan 2 adalah 40 sampai 65. Pada skenario 1 produk independen, semakin besar nilai w yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter w adalah meningkatkan *total profit* gabungan. Untuk skenario 1 produk substitusi, semakin besar nilai w yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter w adalah meningkatkan *total profit* gabungan. Sedangkan pada skenario 2 produk independen, semakin besar nilai w yang digunakan menyebabkan semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter w adalah meningkatkan *total profit* gabungan. Untuk skenario 2 produk substitusi juga memiliki hasil yang sama, yaitu semakin besar nilai w yang digunakan maka semakin besar *total profit* gabungan yang dihasilkan. Sehingga, tren parameter w adalah meningkatkan *total profit* gabungan.

(Halaman ini sengaja dikosongkan)

BAB 6

KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan dari penelitian yang telah dilakukan oleh penulis serta saran dalam mengembangkan penelitian ini.

6.1. Kesimpulan

Adapun kesimpulan pada penelitian ini adalah:

1. Penelitian ini mengembangkan model persediaan dan penetapan harga multi-produk pada *fresh goods* untuk produk independen dan substitusi. Dimana permintaan pada penelitian ini dipengaruhi oleh harga jual, *displayed stock*, dan *expiration date*. Metode yang digunakan dalam penelitian ini adalah metode metaheuristik dengan Algoritma *Artificial Bee Colony*.
2. Skenario 2 memiliki nilai *total profit* yang lebih besar untuk produk independen maupun substitusi. Untuk produk independen *total profit* skenario 2 adalah Rp59.069.280,00 lebih besar jika dibandingkan dengan skenario 1, yaitu Rp59.067.000,00. Selain itu, *total profit* skenario 2 pada produk substitusi juga memiliki nilai yang lebih besar, yaitu Rp2.334.390.290,00 dibandingkan dengan nilai *total profit* skenario 1, yaitu Rp1.341.620.670,00.
3. Analisis sensitivitas dilakukan menggunakan beberapa parameter, yaitu: parameter α (*potential customer*), β (*shelf space efficiency of demand*), λ (*price efficiency of demand*), c (*purchasing cost*), h (*holding cost*), o (*ordering cost*), s (*salvage price*), dan w (*size of shelft space*). Dari percobaan analisis sensitivitas tersebut dapat disimpulkan bahwa semakin tinggi nilai parameter α , β , s , dan w maka semakin tinggi nilai *total profit* gabungan. Sedangkan semakin tinggi nilai parameter c , h , dan o maka semakin kecil nilai *total profit* gabungan. Hal ini dikarenakan parameter tersebut adalah parameter biaya dimana di dalam teori semakin besar biaya yang dikeluarkan maka semakin kecil keuntungan yang didapatkan. Sedangkan untuk parameter λ , semakin

tinggi nilai parameter tersebut maka akan menurunkan *total profit* gabungan produk independen pada skenario 1, produk independen skenario 2, dan produk substitusi skenario 2. Sedangkan untuk produk substitusi skenario 1 mengalami peningkatan *profit*.

6.2. Saran

Dalam melakukan penelitian ini, penulis tentu memiliki kekurangan dan keterbatasan sehingga penelitian ini masih jauh dari kata sempurna dan dapat dikembangkan lebih baik lagi. Beberapa saran yang dapat dipertimbangkan dalam pengembangan penelitian ini kedepannya antara lain: permintaan stokastik, menambahkan *quantity discount*, dan mengizinkan adanya *shortage*.

Selain itu, penulis juga memberikan saran operasional agar model simulasi ini dapat dimanfaatkan untuk membantu manajerial *retail* dalam mengambil keputusan. Model pada penelitian ini memiliki parameter yang dapat diubah sesuai kebutuhan atau situasi yang dihadapi dalam membantu manajerial mengambil keputusan parameter tersebut adalah α (*potential customer*), β (*shelf space efficiency of demand*), λ (*price efficiency of demand*), c (*purchasing cost*), h (*holding cost*), o (*ordering cost*), s (*salvage price*), dan w (*size of shelft space*).

DAFTAR PUSTAKA

- Abad, P. L. (2003). Optimal pricing and lot-sizing under conditions of perishability , finite production and partial backordering and lost sale, *144*, 677–685.
- Banerjee, S., & Agrawal, S. (2017). Inventory model for deteriorating items with freshness and price dependent demand: Optimal discounting and ordering policies. *Applied Mathematical Modelling*, *52*, 53–64. <https://doi.org/10.1016/j.apm.2017.07.020>
- Bhunia, A., & Pareek, S. (2017). An application of Genetic Algorithm and PSO in an inventory model for single deteriorating item with variable demand dependent on marketing strategy and displayed stock level An application of Genetic Algorithm and PSO in an inventory model for single deteriorating item with variable demand dependent on marketing strategy and displayed stock level, (September). <https://doi.org/10.24200/sci.2017.4445>
- Bylka, S. (2019). The consignment stock of inventories under buyer ' s warehouse space limitation, *7543*. <https://doi.org/10.1080/00207543.2019.1605223>
- Chang, H., & Dye, C. (1999). An EOQ model for deteriorating items with time varying demand and partial backlogging, 1176–1182.
- Daniel, J. S. R., & Rajendran, C. (2005). TRANSACTIONS IN OPERATIONAL A simulation-based genetic algorithm for inventory optimization in a serial supply chain, *12*, 101–127.
- Dobson, G., Pinker, E. J., & Yildiz, O. (2017). An EOQ model for perishable goods with age-dependent demand rate. *European Journal of Operational Research*, *257*(1), 84–88. <https://doi.org/10.1016/j.ejor.2016.06.073>
- Feng, L., Chan, Y. L., & Cárdenas-Barrón, L. E. (2017). Pricing and lot-sizing polices for perishable goods when the demand depends on selling price, displayed stocks, and expiration date. *International Journal of Production Economics*, *185*, 11–20. <https://doi.org/10.1016/j.ijpe.2016.12.017>
- Feng, X., Moon, I., & Ryu, K. (2017). Warehouse capacity sharing via transshipment for an integrated two-echelon supply chain. *Transportation Research Part E*, *104*, 17–35. <https://doi.org/10.1016/j.tre.2017.04.014>

- Gilarso, T. (2003). *Pengantar Ilmu Ekonomi Mikro*. Yogyakarta: Kanisius.
- Hidayat, Y. A., & Fauzi, M. R. (2015). Inventory Model for Deteriorating Items with Expired Time in Lost-Sales Probabilistic Demand, 40–46.
- Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm, 8, 687–697. <https://doi.org/10.1016/j.asoc.2007.05.007>
- Kardes, F. R., Cronley, M. L., & Cline, T. W. (2011). *Consumer Behavior*. Mason, OH : South-Western, Cengage Learning.
- Khurana, D., & Chaudhary, R. R. (2018). An order level inventory model for deteriorating stock product and time dependent demand under shortages, 12(3).
- Kırcı, M., Biçer, I., & Seifert, R. W. (2018). Optimal replenishment cycle for perishable items facing demand uncertainty in a two-echelon inventory system. *International Journal of Production Research*, 0(0), 1–15. <https://doi.org/10.1080/00207543.2018.1504244>
- Lu, L., Zhang, J., & Tang, W. (2014). Optimal dynamic pricing and replenishment policy for perishable items with inventory-level-dependent demand. *International Journal of System Science*, 47(6), 1480–1494. <https://doi.org/10.1080/00207721.2014.938784>
- Mishra, P. (2017). iMedPub Journals Optimal Policies for Perishable Items when Demand Depends on Freshness of Displayed Stock and Selling Price Abstract, 1–5. <https://doi.org/10.21767/2394-9988.100060>
- Nahmias, S. (1982). Perishable Inventory Theory: A Review. *Operations Research*, 30(4), 680–708. <https://doi.org/10.1287/opre.30.4.680>
- Optimal Pricing and Lot-Sizing Under Conditions of Perishability and Partial Backordering. (1996), (August 2015).
- Papachristos, S., & Skouri, K. (2003). An inventory model with deteriorating items , quantity discount , pricing and time-dependent partial backlogging, 83, 247–256.
- Rai, S. K., Singh, V., & Vaish, A. (2013). PERISHABLE FOOD INVENTORY MANAGEMENT : A RETAILER ' S PERSPECTIVE, 2(2), 1–10.
- Raouf, A. (1993). On the Constrained Multi-item Single-period Inventory Problem.

- Sana, S. S. (2013). An EOQ model for stochastic demand for limited, (2011).
<https://doi.org/10.1007/s10479-013-1510-5>
- Saracoglu, I., Topaloglu, S., & Keskindurk, T. (2014). Expert Systems with Applications A genetic algorithm approach for multi-product multi-period continuous review inventory models. *EXPERT SYSTEMS WITH APPLICATIONS*, 41(18), 8189–8202.
<https://doi.org/10.1016/j.eswa.2014.07.003>
- Shavandi, H., Mahlooji, H., & Ekram, N. (2012). A constrained multi-product pricing and inventory control problem. *Applied Soft Computing Journal*, 12(8), 2454–2461. <https://doi.org/10.1016/j.asoc.2012.03.036>
- Taleizadeh, A. A., Taghi, S., & Niaki, A. (2010). A genetic algorithm to optimize multiproduct multiconstraint inventory control systems with stochastic replenishment intervals and discount, 311–323.
<https://doi.org/10.1007/s00170-010-2604-8>
- Tiwari, S., Jaggi, C. K., Gupta, M., & Cárdenas-barrón, L. E. (2018). Optimal pricing and lot-sizing policy for supply chain system with deteriorating items under limited storage capacity. *International Journal of Production Economics*. <https://doi.org/10.1016/j.ijpe.2018.03.019>
- Valliathal, M., & Uthayakumar, R. (2011). Optimal pricing and replenishment policies of an EOQ model for non-instantaneous deteriorating items with shortages, 361–371. <https://doi.org/10.1007/s00170-010-2913-y>
- Vrat, P. (1993). EOQ models for perishable items under stock dependent selling rate.
- Vrat, P. (2014). *Materials Management*. <https://doi.org/10.1007/978-81-322-1970-5>
- Wang, S., Lee, W., & Chang, C. (2012). Int . J . Production Economics Modeling the consignment inventory for a deteriorating item while the buyer has warehouse capacity constraint. *Intern. Journal of Production Economics*, 138(2), 284–292. <https://doi.org/10.1016/j.ijpe.2012.03.029>
- Wee, H. (1999). Deteriorating inventory model with quantity discount , pricing and partial backordering, 59, 511–518.
- Wu, J., Chang, C.-T., Cheng, M.-C., Teng, J.-T., & Al-khateeb, F. B. (2016).

Inventorymanagement for fresh produce when the time-varying demand depends on product freshness, stock level and expiration date. *International Journal of Systems Science: Operations & Logistics*, 3(3), 138–147. <https://doi.org/10.1080/23302674.2015.1068880>

Wu, K., Ouyang, L., & Yang, C. (2006). An optimal replenishment policy for non-instantaneous deteriorating items with stock-dependent demand and partial backlogging, *101*, 369–384. <https://doi.org/10.1016/j.ijpe.2005.01.010>

LAMPIRAN

Lampiran 1. Bahasa Pemrograman Algoritma *Artificial Bee Colony* (ABC)

Kalimat yang dimulai dengan “#” merupakan keterangan atau komen

In [1]:

```
# Import library
import pandas as pd # Struktur tabel
import numpy as np # Matrix
import matplotlib.pyplot as plt # Grafik
import random # Bilangan random
import math # operasi matematika
from tqdm import tqdm # Waktu komputasi

%matplotlib inline
```

In [2]:

```
def obj(x, c, h, m, o, s, w, alpha, beta, gamma, product):

    # Decision variable
    Q = [x[0], x[1]] #Order quantity
    P = [x[2], x[3]] #Price
    T = [x[4], x[5]] #Replenishment cycle

    # Inventory t = 0 is Q
    i0 = Q

    # Initiate list
    d = [], [] # Demand
    f = [], [] # Fullfil
    sh = [], [] # Shortage
    bi = [], [] # Beginning inventory
    ei = [], [] # Ending inventory
    rbi = [], [] # Beginning inventory in warehouse
    rei = [], [] # Ending inventory in warehouse
    wbi = [], [] # Beginning inventory in shelf
    wei = [], [] # Ending inventory in shelf
    p = [], [] # Revenue
    tp = [], [] # Total profit
```

```

# Calculate every product
for i in range(2):

    # Calculate every time t
    for t in range(T[i]+1):

        # If t = 0, beginning inventory is i0
        if t == 0:
            bi[i].append(i0[i])
            rbi[i].append(i0[i]-w[i])
            wbi[i].append(w[i])

        # Current beginning inventory
        bit = bi[i][t]
        rbit = rbi[i][t]
        wbit = wbi[i][t]

        # Demand
        # Freshness index
        fr = (m[i] - t)/m[i]

        # Display stock index
        wb = (wbit)**beta[i]

        # Calculate demand
        if t > 0:
            if product == 'independent':
                # Demand
                dt = round(alpha[i] * wb * fr * math.exp(-gamma[i] *
P[i]) / 365)
            elif product == 'substitute':
                # Substitute index
                if i == 0:
                    j = 1
                elif i == 1:
                    j = 0
                # Demand
                dt = round(alpha[i] * wb * fr * math.exp(- gamma[i] *
P[i] + gamma[j] * P[j]) / 365)

```

```

else:
    dt = 0
d[i].append(dt)

# Calculate fullfill and shortage
if bit > dt:
    f[i].append(dt)
    sh[i].append(0)
else:
    f[i].append(bit)
    sh[i].append(dt-bit)
ft = f[i][t]
sht = sh[i][t]

# Calculate ending inventory
if rbit < ft:
    reit = rbit - min(rbit,ft)
    weit = wbit - (ft - rbit)
    eit = reit + weit
    rei[i].append(reit)
    wei[i].append(weit)
    ei[i].append(eit)
else:
    reit = rbit - min(rbit,ft)
    weit = wbit
    eit = reit + weit
    rei[i].append(reit)
    wei[i].append(weit)
    ei[i].append(eit)

# Sales profit
s_p = ft*P[i]

# Holding cost
h_c = eit*h[0]/365

# Profit
pt = s_p - h_c
p[i].append(pt)

```

```

    # Next beginning inventory is current ending inventory
    bi[i].append(eit)
    rbi[i].append(reit)
    wbi[i].append(weit)

    # Unit cost
    u_c = Q[i] * c[i]

    # Salvage revenue
    s_p = ei[i][-1] * s[i]

    # Total profit
    tpp = (sum(p[i][1:]) - u_c - o[i] + s_p)*(365/T[i])
    tp[i].append(tpp)

    # Objective
    obj = tp[0][0] + tp [1][0]

    return obj

```

In [3]:

```

def fitness(obj):

    # Calculate fitness
    if obj >= 0:
        fitness = 1 + 1/(1+(1/(0.001+obj)))
    else:
        fitness = 1/(1+abs(obj))

    return fitness

```

In [4]:

```

def constraint(x, m, w, pm, ps, cap, warehouse):

    # Convert to int
    x = x.astype(int)

    # Change < 0 to 1
    x[x <= 0] = 1

```

```

# T <= m
if x[4] > m[0]:
    x[4] = m[0]
if x[5] > m[1]:
    x[5] = m[1]

# Q > w
if x[0] < w[0]:
    x[0] = w[0] + 1
if x[1] < w[1]:
    x[1] = w[1] + 1

# Mean-std <= price <= mean+std
if x[2] < pm[0] - ps[0]:
    x[2] = pm[0] - ps[0]
elif x[2] > pm[0] + ps[0]:
    x[2] = pm[0] + ps[0]
if x[3] < pm[1] - ps[1]:
    x[3] = pm[1] - ps[1]
elif x[3] > pm[1] + ps[1]:
    x[3] = pm[1] + ps[1]

# Q <= cap
if warehouse == 'single':
    if x[0] > cap[0]:
        x[0] = cap[0]
    if x[1] > cap[1]:
        x[1] = cap[1]
elif warehouse == 'mixed':
    cap_total = sum(cap)
    if x[0] + x[1] > cap_total:
        diff = (x[0] + x[1]) - cap_total
        rand = np.random.random()
        min1 = rand*diff
        min2 = (1-rand)*diff
        if min1 > x[0]:
            to_min2 = min1 - x[0] + 1.5
            min1 = min1 - to_min2
            min2 = min2 + to_min2
        elif min2 > x[1]:

```

```

        to_min1 = min2 - x[1] + 1.5
        min1 = min1 + to_min1
        min2 = min2 - to_min1
        x[0] = int(x[0] - min1)
        x[1] = int(x[1] - min2)
    if x[0] == 0:
        if x[0] + x[1] > cap_total:
            x[0] = 1
            x[1] = x[1] - 1
        else:
            x[0] = 1
    elif x[1] == 0:
        if x[0] + x[1] > cap_total:
            x[1] = 1
            x[0] = x[0] - 1
        else:
            x[1] = 1

    return x

```

In [5]:

```
def initialize(E_bee, D, space):
```

```

    # List to store x
    x = []

    # Initialize each employed bee
    for bee in range(E_bee):

        # Start with empty list:
        xi = []

        # Random every xij with searching space limit
        for di in range(D):
            xij = space[di][0] + random.random()*(space[di][1] -
space[di][0])
            xi.append(xij)

        x.append(xi)

```

```
# Convert to array
x = np.array(x)
```

```
return x
```

In [6]:

```
def find_around(x):
```

```
# Copy x for v
v = x.copy()
```

```
for xi in range(len(x)):
```

```
    # Random k and j
```

```
    k = random.choice([i for i in range(E_bee) if i != xi])
```

```
    j = random.randrange(0, D, 1)
```

```
# Search around
```

```
v[xi][j] = x[xi][j] + random.uniform(-1, 1)*(x[xi][j] - x[k][j])
```

```
return v
```

In [7]:

```
# Parameter
```

```
c = [15.5, 13] # Unit cost per unit
```

```
h = [5, 4] # Holding cost per unit
```

```
m = [15, 11] # Maximum lifetime of product
```

```
o = [8, 7] # Order cost per order
```

```
s = [12, 10] # Salvage price
```

```
w = [50, 50] # Shelf capacity
```

```
pm = [27.5, 25] # Price mean
```

```
ps = [7.5, 10] # Price std
```

```
alpha = [10000, 10000] # Maximum potential customer
```

```
beta = [0.6, 0.5] # Display stock weight
```

```
gamma = [0.1, 0.2] # Price weight
```

```
# Scenario
```

```
product = 'independent' #Product type (independent, substitute)
```

```
warehouse = 'single' #warehouse type (single, mixed)
```

```
cap = [100, 100] #Warehouse capacity
```

```

# Hyper Parameter
n = 5000 #Iteration
P = 30 #Population size
P_ratio = 0.5 #Employed, Onlooker ratio
E_bee = int(P * P_ratio) #Employed bee
O_bee = int(P * P_ratio) #Onlooker bee
limit = 5 #Limit
counter = np.array([0]*E_bee) #Limit counter
D = 6 #Dimension
space = np.array([[0, cap[0]], # Q1 (0-capacity)
                 [0, cap[1]], # Q2 (0-capacity)
                 [pm[0] - ps[0], pm[0] + ps[0]], # P1 [mean-std, mean+std]
                 [pm[1] - ps[1], pm[1] + ps[1]], # P2 [mean-std, mean+std]
                 [0, m[0]], # T1 (0 - max lifetime)
                 [0, m[1]]]) # T2 (0 - max lifetime)
#Inital searching space

```

In [8]:

```

###----- Initialization -----###

```

```

# Initialize solution for each employed bee
x = initialize(E_bee, D, space)

```

```

# Check constraint
x_temp = np.empty((0,D), int)
for xi in x:
    x_temp = np.append(x_temp, [constraint(xi, m, w, pm, ps, cap,
warehouse)], axis=0)
x = x_temp

```

```

# Calculate fitness
fx = []
for xi in x:
    fx.append(fitness(obj(xi, c, h, m, o, s, w, alpha, beta, gamma,
product)))
fx = np.array(fx)

```

```

# Initiate x and f best
f_best = max(fx)
x_best = x[np.where(fx == f_best)][0]

```

```
# Store f_best in all iteration
f_best_list = []
```

In [9]:

```
###----- Finding solution -----###
```

```
# Loop in iteration
for it in tqdm(range(n)):
```

```
    ### Employee bee phase ###
```

```
    # Find around from current solution
```

```
    v = find_around(x)
```

```
    # Check constraint
```

```
    v_temp = np.empty((0,D), int)
```

```
    for vi in v:
```

```
        v_temp = np.append(v_temp, [constraint(vi, m, w, pm, ps, cap,
warehouse)], axis=0)
```

```
    v = v_temp
```

```
    # Calculate fitness
```

```
    fv = []
```

```
    for vi in v:
```

```
        fv.append(fitness(obj(vi, c, h, m, o, s, w, alpha, beta, gamma,
product)))
```

```
    fv = np.array(fv)
```

```
    # Store new solution with greedy selection and update counter
```

```
    for i in range(E_bee):
```

```
        if fv[i] > fx[i]:
```

```
            fx[i] = fv[i]
```

```
            x[i] = v[i]
```

```
            counter[i] = 0
```

```
        else:
```

```
            counter[i] += 1
```

```
    # Update best solution
```

```
    if max(fv) > f_best:
```

```
        f_best = max(fv)
```

```

x_best = x[np.where(fx == f_best)][0]

### Onlooker Bee Phase ###
# Calculate probability based on employed bee fitness
p = fx/fx.sum()
p = p.cumsum()

# Roulette wheel selection
for i in range(O_bee):

    # Random number
    rand = random.random()

    # Roulette wheel selection
    for pi in range(len(p)):
        if rand < p[pi]:
            selected = pi
            break

    # Onlooker find new solution
    oi = find_around(x)[selected]

    # Check constraint
    oi = constraint(oi, m, w, pm, ps, cap, warehouse)

    # Fitness of onlooker
    foi = fitness(obj(oi, c, h, m, o, s, w, alpha, beta, gamma, product))

    # Compare new solution by onlooker with greedy selection
    if foi > fx[selected]:
        fx[selected] = foi
        x[selected] = oi
        counter[selected] = 0
    else:
        counter[selected] += 1

# Update best solution
if max(fx) > f_best:
    f_best = max(fx)
    x_best = x[np.where(fx == f_best)][0]

```

```

#### Scout bee phase ####
# Check if any abandoned solution
if sum(counter > limit) > 0:
    # Scout bee find new solution for abandoned solution
    for i in np.where(counter > limit)[0]:
        # Generate new random solution
        xi_new = []
        for di in range(D):
            xij = space[di][0] + random.random()*(space[di][1] -
space[di][0])
            xi_new.append(xij)
        xi_new = np.array(xi_new)

        # Check non zero constraint
        xi_new = constraint(xi_new, m, w, pm, ps, cap, warehouse)

        # Replace abandoned solution
        x[i] = xi_new
        fx[i] = fitness(obj(xi_new, c, h, m, o, s, w, alpha, beta, gamma,
product))

        # Refresh counter
        counter[i] = 0

    # Update best solution
    if max(fx) > f_best:
        f_best = max(fx)
        x_best = x[np.where(fx == f_best)][0]
    else:
        pass

# Store f best
f_best_list.append(f_best)

```