

BAB IV

PENGEMBANGAN MODEL

Bab ini menjelaskan mengenai deskripsi sistem, proses pengembangan model, validasi model, dan pengolahan data berupa contoh numerik untuk model.

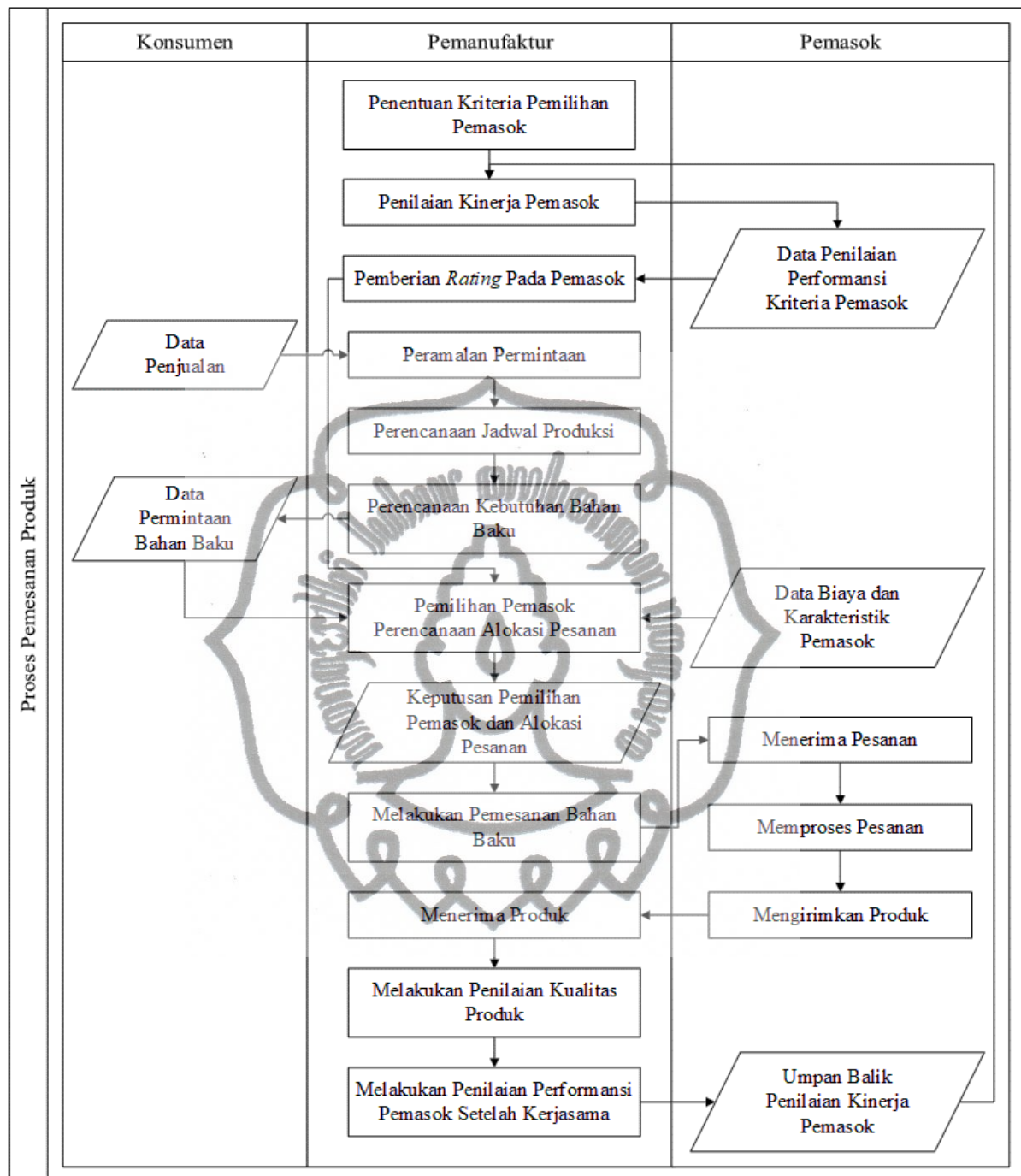
4.1 DESKRIPSI DAN KARAKTERISTIK SISTEM

Sistem yang dimodelkan dalam penelitian ini merupakan sistem pemilihan pemasok dan alokasi pesanan yang mempertimbangkan kriteria keberlanjutan dan ketahanan terhadap resiko disrupsi untuk mendukung perusahaan dalam menghadapi kompetisi global yang penuh dengan ketidakpastian.

Pihak yang dilibatkan secara langsung dalam sistem ini adalah pamanufaktur sebagai *buyer* dan pemasok sebagai *vendor*. Terdapat juga pihak konsumen sebagai penyedia data terkait permintaan penjualan. Dari data penjualan ini, pamanufaktur dapat melakukan peramalan permintaan produk, perencanaan jadwal produksi, dan perencanaan kebutuhan bahan baku sehingga dapat diperoleh data kebutuhan bahan baku untuk yang dibutuhkan dalam proses produksi.

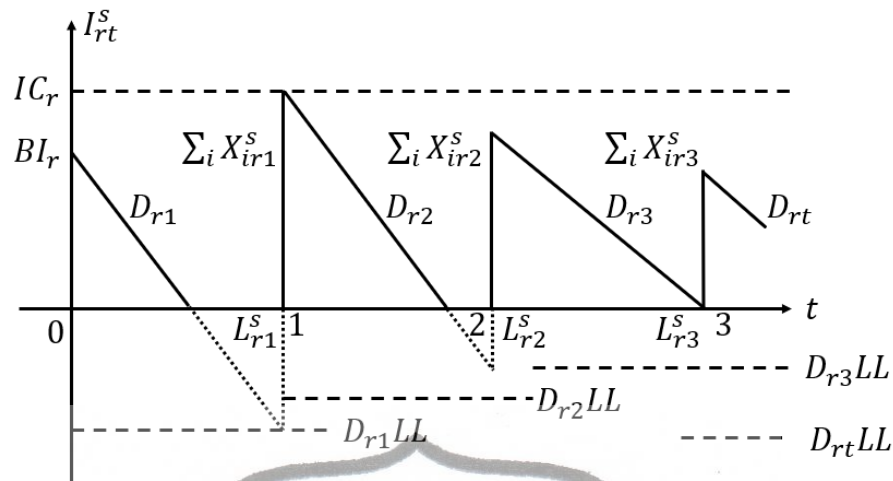
Di sisi lain, ahli dan pemangku kepentingan pihak pamanufaktur dapat melakukan perumusan kriteria yang akan digunakan untuk melakukan evaluasi pemasok. Pemasok dapat menyediakan data yang dibutuhkan pamanufaktur untuk melakukan penilaian kinerja. Penilaian ini juga dapat didukung dari umpan balik yang didapatkan perusahaan dari aktivitas kerjasama yang telah dilakukan.

Dari hasil penilaian kinerja atau performansi didapatkan bobot pada masing-masing kandidat pemasok yang selanjutnya digunakan dalam tahap pemilihan pemasok dan alokasi pesanan. Tahapan ini juga didukung oleh data-data biaya dan karakteristik produksi dari pemasok. Setelah pemasok sudah ditetapkan dan alokasi pesanan sudah direncanakan, maka dilakukan proses pemesanan. Pemasok akan menerima dan memproses pesanan tersebut dan selanjutnya mengirimkan produk tersebut ke pamanufaktur. Selanjutnya, pamanufaktur dapat melakukan penilaian terhadap hasil produk yang dikirim dan kinerja pemasok selama kerjasama dilakukan. Informasi ini kemudian dapat digunakan sebagai umpan balik saat melakukan penilaian kinerja pemasok selanjutnya.



Gambar 4.1 Diagram Sistem Pemilihan Pemasok dan Alokasi Pesanan

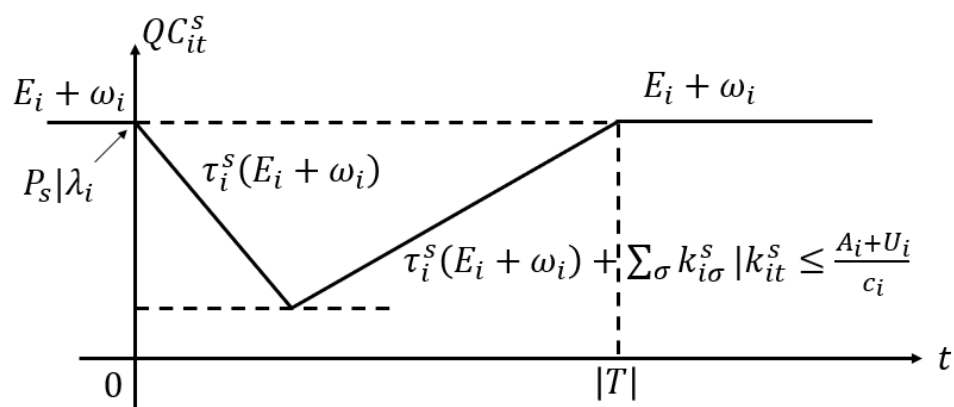
Dalam sistem yang dimodelkan, digunakan prinsip tingkat persediaan pamanufaktur dengan memperbolehkan terjadinya kehilangan penjualan dalam tingkat tertentu untuk setiap produk. Karakteristik tingkat persediaan dalam sistem yang dimodelkan dijelaskan pada grafik berikut.



Gambar 4.2 Tingkat Persediaan dengan Kehilangan Penjualan

Tingkat persediaan pada awal periode adalah sejumlah $B I_r$. Penurunan pada tingkat persediaan (I_{rt}^s) dalam jangka waktu tertentu (t) dipengaruhi oleh laju permintaan (D_{rt}). Kehilangan penjualan maksimal sejumlah LL dari permintaan diperbolehkan untuk terjadi dan pemesanan sejumlah X_{irt}^s dapat dilakukan untuk memenuhi permintaan. Jumlah pemesanan ini tidak boleh lebih besar daripada kapasitas gudang (IC_r).

Dalam sistem yang dimodelkan, terdapat penurunan terhadap tingkat kapasitas pemasok saat disrupsi terjadi. Lalu, sebelum akhir periode kapasitas harus terpulihkan kembali pada tingkat kapasitas awal sebelum terjadinya disrupsi. Pengaruh kejadian disrupsi terhadap tingkat kapasitas produksi pemasok dijelaskan pada grafik berikut.



Gambar 4.3 Tingkat Kapasitas Pemasok Selama Disrupsi

Mulai dari awal disrupsi, terjadi penurunan kapasitas produksi akibat terjadinya gangguan. Kapasitas produksi ini terdiri dari kapasitas produksi primer (E_i) dan kapasitas tambahan (ω_i). Penurunan kapasitas terjadi hingga tingkat kapasitas berada pada τ_i^S dari keseluruhan tingkat kapasitas produksi awal. Penurunan kapasitas ini disebabkan oleh kerentanan dalam sistem terhadap kejadian disrupsi. Selanjutnya terjadi peningkatan kapasitas produksi karena proses pemulihan. Kapasitas produksi yang terpulihkan (k_{it}^S) ini bergantung kepada sumber daya yang tersedia, berupa kapasitas restoratif (A_i) dan kapasitas restoratif tambahan (U_i). Kapasitas produksi memerlukan c_i banyak sumber daya yang berupa kapasitas restoratif agar dapat terpulihkan.

4.2 FORMULASI MODEL

Bagian ini menjelaskan mengenai formulasi model yang dikembangkan meliputi kerangka kerja model, kriteria keberlanjutan, prosedur penilaian pemasok, komponen model, *influence diagram*, asumsi model, fungsi tujuan, dan batasan model.

4.2.1 Kerangka Kerja Model

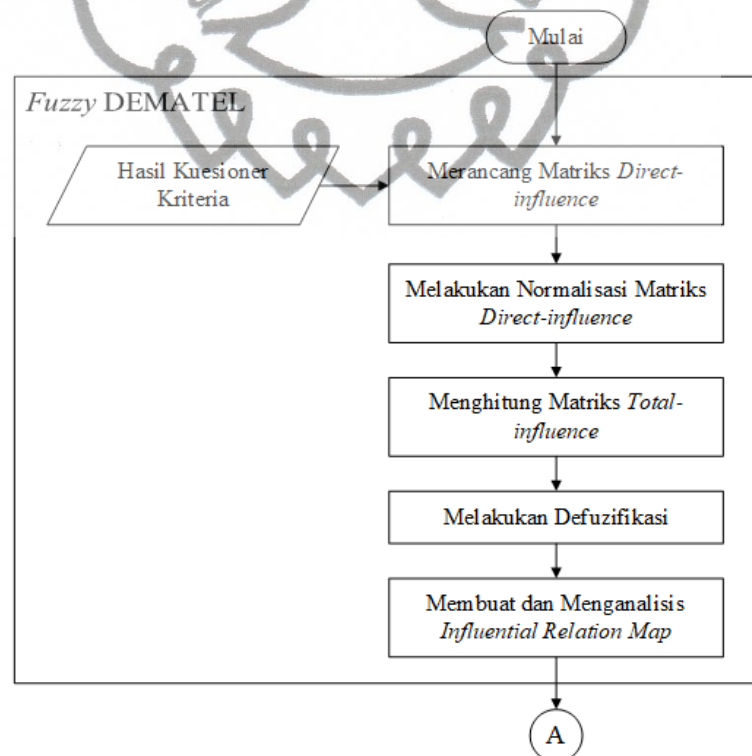
Pada penelitian ini model permasalahan pemilihan pemasok dan alokasi pesanan atau *Supplier Selection and Order Allocation Problem* (SS-OAP) didasarkan pada beberapa penelitian, yaitu: Goren (2018), Hosseini, dkk. (2019), dan Hsu, dkk (2012). Dalam penelitian Goren (2018) untuk menentukan bobot pemasok digunakan *framework* (kerangka kerja) model melalui dua tahapan MADM (*Multi Attribute Decision Making*) terdiri dari DEMATEL (*Decision Making Trial and Evaluation Laboratory*) dan *Taguchi Loss Function*. Dalam model yang dikembangkan, digunakan tahapan DANP (DEMATEL-based *Analytic Network Process*) diantara tahapan DEMATEL dan *Taguchi Loss Function*. Hal ini dilakukan untuk mengevaluasi ketergantungan kriteria dengan dimensinya. Tahapan DANP ini didasarkan pada penelitian Hsu, dkk. (2012).

Dalam penelitian Hosseini, dkk. (2019), digunakan MO-MILP (*Multiple Objective Mixed Integer Linear Programming*) untuk menentukan pemasok dan alokasi pesanan dengan mempertimbangkan disrupsi pada kapasitas produksi. Dalam model yang dikembangkan, variabel keputusan kuantitas pesanan dan kuantitas kehilangan penjualan merupakan bilangan bulat daripada sebagai fraksi

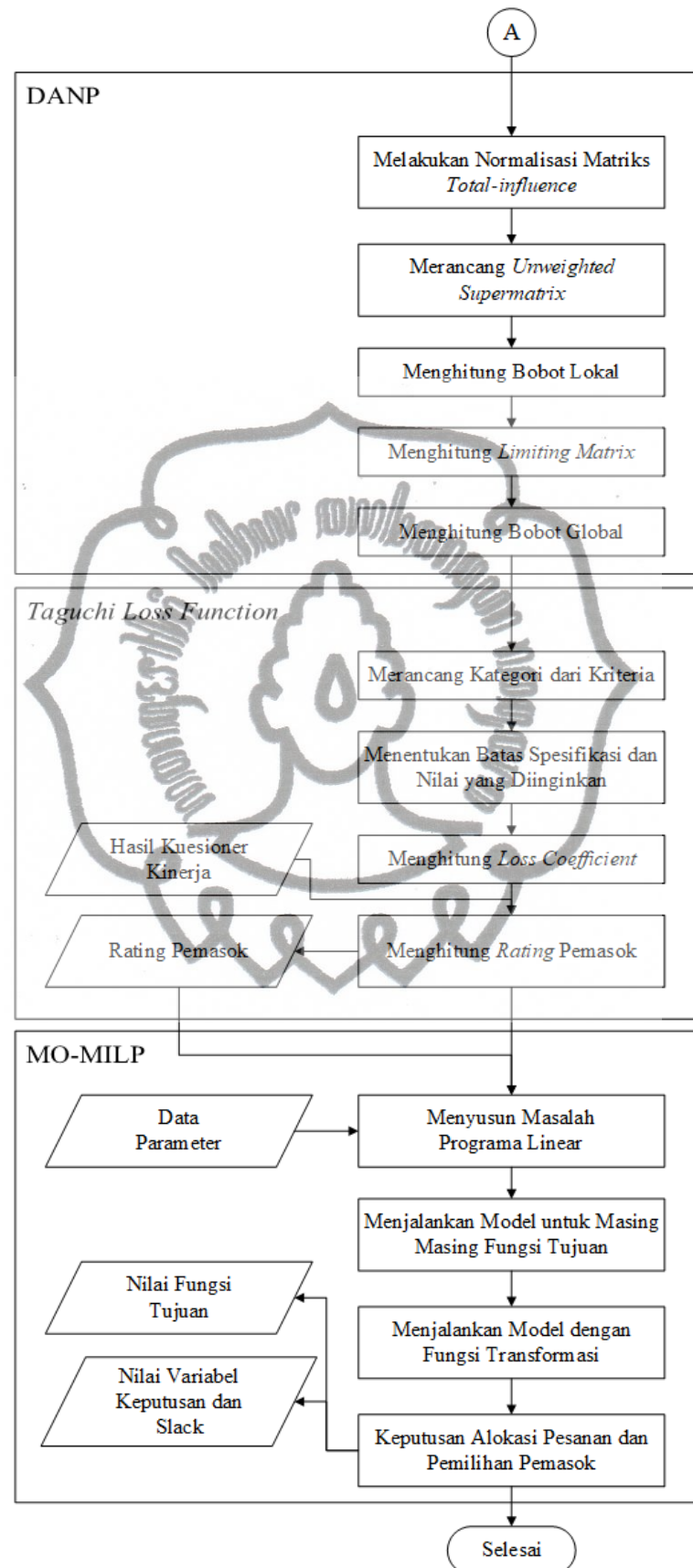
dari permintaan (bilangan real). Dalam model yang dikembangkan, terdapat pula batasan waktu tunggu dan tingkat persediaan.

Sehingga, kerangka kerja model SS-OAP yang dikembangkan terdiri dari empat tahap pengolahan data, meliputi: *Fuzzy DEMATEL*, DANP, *Taguchi Loss Function*, dan MO-MILP. DEMATEL digunakan untuk mengolah penilaian keterkaitan atau pengaruh antar kriteria serta membuat matriks pengaruh langsung, matriks pengaruh total, dan IRM (*Influential Relation Map*). DANP digunakan untuk memperoleh bobot lokal dan IRM untuk setiap kriteria dalam dimensi serta bobot global melalui *limiting matrix*. *Taguchi Loss Function* digunakan untuk memperoleh nilai *rating* untuk setiap pemasok melalui perhitungan kerugian total dari keseluruhan kriteria untuk setiap alternatif. MO-MILP digunakan untuk memperoleh keputusan pemilihan pemasok dan alokasi pesanan yang optimal berdasarkan beberapa batasan dan fungsi tujuan yang digunakan.

Kerangka kerja dari model SS-OAP yang dikembangkan adalah sebagai berikut.



Gambar 4.4 Kerangka Kerja Model



Gambar 4.4 Kerangka Kerja Model (lanjutan)

4.2.2 Kriteria Keberlanjutan

Kriteria yang digunakan dalam penelitian ini didasarkan pada studi yang dilakukan oleh Hacer G. Goren (2018). Kriteria yang digunakan untuk penilaian *rating* pemasok adalah kriteria keberlanjutan atau *triple bottom line*. Kriteria ini dikelompokkan dalam tiga kategori, yaitu dimensi ekonomi, sosial dan lingkungan. Kriteria ini ditentukan berdasarkan tinjauan literatur dan telah divalidasi oleh pakar perusahaan (Goren, 2018).

Tabel berikut memperlihatkan dimensi dan kriteria yang dipertimbangkan dalam penilaian *rating* pemasok pada model yang dikembangkan.

Tabel 4.1 Dimensi dan Kriteria Keberlanjutan

Dimensi	Simbol	Kriteria	Simbol
Ekonomi	D_1	Harga	C_1
		Produktivitas	C_2
		Kapasitas	C_3
		Kontinuitas	C_4
		Waktu Tunggu	C_5
		Kualitas	C_6
		Teknologi Produksi	C_7
		Responsivitas	C_8
Sosial	D_2	OHSMS	C_9
		Kegiatan Pendukung	C_{10}
Lingkungan	D_3	EMS	C_{11}
		Desain Produk Ramah Lingkungan	C_{12}
		Konsumsi Sumber Daya	C_{13}

Dimensi ekonomi mencakup fitur ekonomi yang dipertimbangkan dari pemasok. Dimensi ini terdiri dari 8 kriteria, yaitu: harga, produktivitas, kapasitas, kontinuitas, waktu tunggu, kualitas, teknologi produksi, dan responsivitas. Penjelasan dari masing-masing kriteria ini adalah sebagai berikut.

1. Harga

Harga adalah kuantitas pembayaran yang diberikan oleh pihak pamanufaktur kepada pemasok sebagai imbalan atas satu unit barang yang didapatkan. Karena pamanufaktur pasti akan lebih tertarik untuk membeli produk dengan harga yang lebih rendah untuk meminimalkan biaya

keseluruhan, kriteria ini merupakan kriteria pertama yang dipertimbangkan ketika melakukan evaluasi pemasok.

2. Produktivitas

Produktivitas berkaitan dengan persentase barang atau produk jadi yang diperoleh dari jumlah bahan baku atau material yang digunakan. Kriteria ini mengandung resiko akibat beberapa bahan baku didapatkan dari sumber daya alam. Oleh karena itu, tidak mungkin untuk mendapatkan efisiensi sebesar 100% dari proses, seperti pada proses pengilangan minyak dan pengolahan hasil tambang. Karena memiliki ketergantungan dengan sumber alam, kriteria ini biasanya sangat bergantung pada situs atau tempat pemasok berada.

3. Kapasitas

Kapasitas produksi adalah volume produk yang dapat dihasilkan oleh pemasok dalam periode waktu tertentu dengan menggunakan sumber daya yang dimiliki pada saat periode tersebut berlangsung. Untuk memenuhi kebutuhan pamanufaktur, pemasok harus memiliki kapasitas yang cukup. Kriteria ini mengandung resiko tersendiri yaitu kemungkinan pemasok tidak dapat memenuhi persyaratan perusahaan.

4. Kontinuitas

Kriteria ini mendefinisikan jenis hubungan antara pemasok dan perusahaan. Hubungan jangka panjang lebih disukai oleh pamanufaktur, karena hal ini dapat menjamin keberlanjutan pemesanan, kemudahan pemesanan berulang, dan ketersediaan bahan baku produksi bagi pamanufaktur.

5. Waktu Tunggu

Waktu tunggu adalah rentang waktu antara saat pesanan diajukan oleh pamanufaktur kepada pemasok dan saat produk diterima oleh pamanufaktur. Pamanufaktur cenderung memilih untuk bekerja dengan pemasok yang memiliki waktu tunggu yang lebih singkat.

6. Kualitas

Kualitas adalah kesesuaian komitmen yang dibuat pemasok terhadap pamanufaktur mengenai kinerja produk. Selalu terdapat resiko di mana produk yang dibeli mungkin tidak cocok dengan persyaratan pamanufaktur.

Kualitas yang disyaratkan oleh pamanufaktur dapat berupa kesesuaian toleransi dimensi, kesesuaian material bahan, dan lain sebagainya.

7. Teknologi Produksi

Teknologi produksi pemasok sangat penting untuk dipertimbangkan dalam proses evaluasi. Teknologi terbaru memungkinkan pemasok untuk menanggapi pesanan pamanufaktur dengan lebih cepat. Selain itu, pemasok yang memiliki teknologi terbaru biasanya lebih tangkas (*agile*) dan dapat menerima pemesanan produk dengan spesifikasi unik. Oleh karena itu, perusahaan akan lebih memilih bekerja dengan pemasok yang memiliki teknologi baru daripada bekerja dengan pemasok yang memiliki teknologi lama.

8. Responsivitas

Responsivitas atau daya tanggap berkaitan dengan perilaku pemasok mengenai layanan pemesanan. Pamanufaktur biasanya ingin mengetahui pembaruan tentang status pesanan dan mendapatkan umpan balik sesegera mungkin.

Dimensi sosial berfokus pada kondisi tempat kerja dan kemajuan karyawan melalui kegiatan yang mendukung. Penjelasan dari kriteria untuk dimensi sosial adalah sebagai berikut.

1. *Occupational Health and Safety Management System (OHSMS)*

Kriteria ini terkait dengan resiko kesehatan dan keselamatan. Dengan sistem manajemen kesehatan dan keselamatan kerja, organisasi dapat menyediakan tempat kerja yang lebih aman bagi karyawan mereka dengan mengelola resiko kesehatan dan keselamatan. Adanya sertifikasi ISO 45001:2018 pada pemasok dapat dijadikan pertimbangan dalam menilai kriteria ini.

2. Kegiatan Pendukung

Kegiatan pendukung mencakup kegiatan yang meningkatkan kepuasan pekerja seperti perhatian pada masalah agama dan budaya di tempat kerja, upah, insentif, serta kegiatan pelatihan dan pembelajaran.

Dimensi lingkungan mencakup kriteria yang terkait dengan permasalahan dan kegiatan pelestarian lingkungan alam. Dimensi ini terdiri dari 3 kriteria, yaitu: sistem manajemen lingkungan, desain produk ramah lingkungan, dan konsumsi sumber daya. Berikut adalah penjelasan dari masing-masing kriteria.

1. *Environmental Management System (EMS)*

Pertimbangan dalam kriteria ini mencakup sertifikasi lingkungan seperti ISO 14001:2015, kebijakan lingkungan, kegiatan perencanaan dan pengecekan lingkungan pabrik, pengendalian dan penekanan emisi produksi dan transportasi, serta pencegahan polusi dan pencemaran lingkungan sekitar dari limbah pabrik.

2. *Desain Produk Ramah Lingkungan*

Kriteria ini menilai aspek ramah lingkungan dari desain produk. Di mana produk didesain untuk mengurangi konsumsi bahan baku dan energi. Selain itu, produk dapat didesain sesuai dengan prinsip 3R, yaitu dapat digunakan kembali (*reuse*), didaur ulang (*recycle*), dan dipulihkan (*recovery*).

3. *Konsumsi Sumber Daya*

Kriteria ini terkait dengan efisiensi dan perilaku konsumsi sumber daya dalam bentuk bahan baku, energi, dan air yang digunakan dalam produksi. Penggunaan sumber energi alternatif seperti panel surya untuk keperluan produksi juga dapat dipertimbangkan.

4.2.3 Prosedur Penilaian Pemasok

Prosedur penilaian pemasok terdiri dari tiga tahap, yaitu: *Fuzzy DEMATEL*, *DANP*, serta *Taguchi Loss Function*. Berikut adalah penjelasan dari prosedur penilaian pemasok.

- A. *Fuzzy DEMATEL (Decision Making Trial and Evaluation Laboratory)*

Tahapan DEMATEL terdiri dari proses pengumpulan data pengaruh kriteria hingga mendapatkan IRM (*Influential Relation Map*). Tahapan pengolahan metode *fuzzy DEMATEL* pada kerangka kerja ini didasarkan pada penelitian Goren (2018) dan Si, dkk. (2018).

1. Pembuatan Matriks Pengaruh Langsung

Tahapan pertama dalam pengolahan *fuzzy* DEMATEL adalah membuat penilaian hubungan pengaruh langsung antar setiap kriteria untuk membuat *direct-influential matrix*. Penilaian ini dapat dilakukan oleh sekumpulan responden ahli melalui kuesioner serta melalui kajian kelompok secara ekstensif. Dalam sistem *fuzzy* penilaian dilakukan dengan menggunakan variable linguistik. Penilaian dalam variable linguistik ini selanjutnya akan diinterpretasikan dalam bentuk bilangan *fuzzy* menggunakan skala linguistik *fuzzy*. Skala linguistik *fuzzy* yang digunakan dalam kerangka kerja model ini didasarkan pada penelitian Li (1999) yang menggunakan bilangan *fuzzy* segitiga.

Tabel 4.2 Skala Linguistik *Fuzzy* (Li, 1999)

Variabel Linguistik	Definisi	Nilai Crisp	Bilangan Fuzzy
N	Tidak terdapat pengaruh (<i>No influence</i>)	0	(0, 0, 1/4)
VL	Pengaruh sangat lemah (<i>Very low influence</i>)	1	(0, 1/4, 1/2)
L	Pengaruh lemah (<i>Low influence</i>)	2	(1/4, 1/2, 3/4)
H	Pengaruh kuat (<i>High influence</i>)	3	(1/2, 3/4, 1)
VH	Pengaruh sangat kuat (<i>Very high influence</i>)	4	(3/4, 1, 1)

Hasil penilaian dalam bentuk bilangan *fuzzy* ini digunakan untuk membentuk matriks pengaruh langsung atau *direct-influential matrix*. Matriks pengaruh langsung individual $\tilde{Z}_k = [\tilde{z}_{ij}^k]_{n \times n}$ didapatkan dari penilaian sekumpulan responden ahli $E = \{E_k | k \in 1, 2, 3, \dots, s\}$. Dimana $\tilde{z}_{ij}^k = (z_{(l)ij}^k, z_{(m)ij}^k, z_{(r)ij}^k)$ merupakan hasil penilaian ahli E_k mengenai derajat pengaruh antara kriteria C_i dan C_j . Selanjutnya matriks pengaruh langsung individual ini diaggregasi sehingga mendapatkan matriks pengaruh langsung kelompok $\tilde{Z} = [\tilde{z}_{ij}]_{n \times n}$.

Aggregasi matriks pengaruh langsung individual dapat dilakukan dengan menggunakan persamaan berikut.

$$\tilde{z}_{ij} = \frac{1}{s} \sum_{k=1}^s \tilde{z}_{ij}^k \quad (4.1)$$

$$(z_{(l)ij}, z_{(m)ij}, z_{(r)ij}) = \left(\frac{1}{s} \sum_{k=1}^s z_{(l)ij}^k, \frac{1}{s} \sum_{k=1}^s z_{(m)ij}^k, \frac{1}{s} \sum_{k=1}^s z_{(r)ij}^k \right) \quad (4.2)$$

2. Normalisasi Matriks Pengaruh Langsung

Matriks pengaruh langsung ternormalisasi $\tilde{X} = [\tilde{x}_{ij}]_{n \times n}$ didapatkan melalui rumus berikut.

$$\tilde{X} = \frac{\tilde{Z}}{\lambda} \quad (4.3)$$

$$\lambda = \max_{1 \leq i \leq n} \left(\sum_{j=1}^n z_{(r)ij} \right) \quad (4.4)$$

Dengan asumsi bahwa setidaknya satu i memenuhi

$$\sum_{j=1}^n z_{(r)ij} < \lambda \quad (4.5)$$

3. Perhitungan Matriks Pengaruh Total

Matriks pengaruh total atau *total-influential matrix* $\tilde{T}_C = [\tilde{t}_{ij}]_{n \times n}$ dapat diperoleh melalui rumus berikut.

$$\tilde{T}_C = \lim_{h \rightarrow \infty} (\tilde{X}^1 + \tilde{X}^2 + \tilde{X}^3 + \dots + \tilde{X}^h) = \tilde{X}(\mathbf{I} - \tilde{X})^{-1} \quad (4.6)$$

Ketika,

$$\lim_{h \rightarrow \infty} \tilde{X}^h = \emptyset \quad (4.7)$$

Dalam hal ini, $\tilde{t}_{ij} = (t_{(l)ij}, t_{(m)ij}, t_{(r)ij})$ dapat dihitung dengan cara:

$$\mathbf{T}_l = [t_{(l)ij}]_{n \times n} = \mathbf{X}_l(\mathbf{I} - \mathbf{X}_l)^{-1} \quad (4.8)$$

$$\mathbf{T}_m = [t_{(m)ij}]_{n \times n} = \mathbf{X}_m(\mathbf{I} - \mathbf{X}_m)^{-1} \quad (4.9)$$

$$\mathbf{T}_r = [t_{(r)ij}]_{n \times n} = \mathbf{X}_r(\mathbf{I} - \mathbf{X}_r)^{-1} \quad (4.10)$$

Di mana, $\mathbf{X}_l = [x_{(l)ij}]_{n \times n}$, $\mathbf{X}_m = [x_{(m)ij}]_{n \times n}$, $\mathbf{X}_r = [x_{(r)ij}]_{n \times n}$, dan \mathbf{I} adalah matriks identitas. Elemen dari bilangan *fuzzy* segitiga pada

matriks \tilde{T}_C dibagi kedalam tiga matriks yaitu T_l , T_m , dan T_r . Di mana $T_l < T_m < T_r$ dan $x_{(l)ij} < x_{(m)ij} < x_{(r)ij}$ untuk setiap $i, j \in \{1, 2, 3, \dots, n\}$.

4. Defuzzifikasi

Defuzzifikasi atau *defuzzification* matriks pengaruh total $\tilde{T}_C = [\tilde{t}_{ij}]_{n \times n}$ dilakukan dengan menggunakan metode CFCS (*Converting Fuzzy Data into Crisp Scores*) oleh Opricovic dan Tzeng (2002). Bila diketahui bahwa elemen bilangan *fuzzy* segitiga dari matriks pengaruh total terdiri dari $\tilde{t}_{ij} = (t_{(l)ij}, t_{(m)ij}, t_{(r)ij})$, berikut adalah langkah-langkah defuzzifikasi dari \tilde{t}_{ij} .

Pertama, dilakukan perhitungan untuk menentukan normalisasi elemen bilangan *fuzzy* $d_{(l)j}$, $d_{(m)j}$, dan $d_{(r)j}$ untuk setiap $j = 1, \dots, n$.

$$d_{(l)j} = (t_{(l)ij} - t_{(l)i}^{\min}) / \Delta_{\min}^{\max} \quad (4.11)$$

$$d_{(m)j} = (t_{(m)ij} - t_{(l)i}^{\min}) / \Delta_{\min}^{\max} \quad (4.12)$$

$$d_{(r)j} = (t_{(r)ij} - t_{(l)i}^{\min}) / \Delta_{\min}^{\max} \quad (4.13)$$

Di mana,

$$t_{(r)i}^{\max} = \max_{1 \leq j \leq n} (t_{(r)ij}^{\max}) \quad (4.14)$$

$$t_{(l)i}^{\min} = \max_{1 \leq j \leq n} (t_{(l)ij}^{\min}) \quad (4.15)$$

$$\Delta_{\min}^{\max} = t_{(r)i}^{\max} - t_{(l)i}^{\min} \quad (4.16)$$

Kemudian, hitung nilai normalisasi sisi kiri d_j^{lhs} dan sisi kanan d_j^{rhs} untuk setiap $j = 1, \dots, n$.

$$d_j^{lhs} = d_{(m)j} / (1 + d_{(m)j} - d_{(l)j}) \quad (4.17)$$

$$d_j^{rhs} = d_{(r)j} / (1 + d_{(r)j} - d_{(m)j}) \quad (4.18)$$

Lalu, hitung total nilai *crisp* ternormalisasi untuk setiap $j = 1, \dots, n$.

$$d_j^{crs} = \frac{d_j^{lhs}(1 - d_j^{lhs}) + d_j^{rhs}d_j^{rhs}}{1 - d_j^{lhs} + d_j^{rhs}} \quad (4.19)$$

Terakhir, hitung nilai *crisp* dari matriks pengaruh total t_{ij} .

$$t_{ij} = t_{(l)i}^{\min} + d_j^{crs} \Delta_{\min}^{\max} \quad (4.20)$$

Sehingga, diperoleh matriks pengaruh total dalam bentuk bilangan *crisp* $T_c = [t_{ij}]_{n \times n}$.

5. Perancangan IRM (*Influential Relation Map*)

Pada langkah ini, dilakukan perhitungan vektor \vec{R} dan \vec{C} , yang merupakan penjumlahan baris dan kolom dari matriks pengaruh total.

$$\vec{R} = [r_i]_{n \times 1} = \left[\sum_{j=1}^n t_{ij} \right]_{n \times 1} \quad (4.21)$$

$$\vec{C} = [c_j]_{1 \times n} = \left[\sum_{i=1}^n t_{ij} \right]'_{1 \times n} \quad (4.22)$$

Dari nilai vektor tersebut dapat dirancang sebuah IRM dengan $\vec{R} + \vec{C}$ sebagai sumbu horizontal yang menandakan derajat *prominence* dan $\vec{R} - \vec{C}$ sebagai sumbu vertikal yang menandakan derajat relasi.

B. DANP (*DEMATEL-based Analytic Network Process*)

Tahapan metode DANP terdiri dari proses pembuatan supermatriks tanpa bobot sampai pada mendapatkan bobot global untuk setiap kriteria. Tahap pengolahan metode DANP dalam kerangka kerja ini didasarkan pada penelitian Hsu, dkk. (2012).

1. Normalisasi Matriks Pengaruh Total

Tahapan ini dimulai dengan menyusun matriks pengaruh total dimensi $T_D = [t_{uv}^D]_{m \times m}$ dari matriks pengaruh total kriteria $T_c = [t_{ij}]_{n \times n}$. Suatu dimensi D_u dalam himpunan $D = \{D_u \text{ atau } D_v | u, v \in 1, 2, 3, \dots, m\}$ terdiri dari beberapa kriteria C_{ui} dalam himpunan $C = \{C_i \text{ atau } C_j | i, j \in 1, 2, 3, \dots, n\}$ di mana kriteria tersebut hanya termasuk dalam satu dimensi.

$$D_u = \{C_{ui} \text{ atau } C_{uj} | i, j \in 1, 2, 3, \dots, n'_u \text{ dan } n'_u \in 1, 2, 3, \dots, n\} \quad (4.23)$$

$$T_C = \begin{matrix} & \begin{matrix} D_1 & \dots & D_v & \dots & D_m \end{matrix} \\ \begin{matrix} D_1 \\ \vdots \\ D_u \\ \vdots \\ D_m \end{matrix} & \begin{matrix} C_{11} & \dots & C_{1n'_1} & \dots & C_{v1} & \dots & C_{vn'_v} & \dots & C_{m1} & \dots & C_{mn} \end{matrix} \end{matrix} \left[\begin{matrix} T_C^{11} & \dots & T_C^{1v} & \dots & T_C^{1m} \\ \vdots & & \vdots & & \vdots \\ T_C^{u1} & \dots & T_C^{uv} & \dots & T_C^{um} \\ \vdots & & \vdots & & \vdots \\ T_C^{m1} & \dots & T_C^{mv} & \dots & T_C^{mm} \end{matrix} \right] \quad (4.24)$$

$$T_C^{uv} = \begin{matrix} C_{u1} \\ \vdots \\ C_{ui} \\ \vdots \\ C_{un'_u} \end{matrix} \left[\begin{matrix} C_{v1} & \dots & C_{vj} & \dots & C_{vn'_v} \\ t_{11}^{uv} & \dots & t_{1j}^{uv} & \dots & t_{1n'_v}^{uv} \\ \vdots & & \vdots & & \vdots \\ t_{i1}^{uv} & \dots & t_{ij}^{uv} & \dots & t_{in'_v}^{uv} \\ \vdots & & \vdots & & \vdots \\ t_{n'_u1}^{uv} & \dots & t_{n'_uj}^{uv} & \dots & t_{n'_un'_v}^{uv} \end{matrix} \right] \quad (4.25)$$

Berikut adalah perhitungan untuk menentukan matriks pengaruh total dimensi $T_D = [t_{uv}^D]_{m \times m}$.

$$t_{uv}^D = \frac{1}{n'_u \times n'_v} \sum_{i=1}^{n'_u} \sum_{j=1}^{n'_v} t_{ij}^{uv} \quad (4.26)$$

Di mana t_{ij}^{uv} merupakan elemen dari matriks T_C^{uv} yang merupakan matriks bagian dari T_C .

Lalu, dilakukan perhitungan untuk normalisasi matriks pengaruh total kriteria berdasarkan dimensi. Matriks pengaruh total kriteria ternormalisasi $T_C^\alpha = [t_{ij}^\alpha]_{n \times n}$ dapat dihitung menggunakan rumus berikut.

$$t_{ij}^\alpha = t_{ij} / \sum_{j=1}^{n'_v} t_{ij}^{uv} \quad (4.27)$$

Elemen t_{ij}^α merupakan hasil pembagian elemen t_{ij} dengan penjumlahan baris elemen-elemen kriteria pada dimensi yang sesuai.

Selanjutnya, dilakukan perhitungan untuk normalisasi matriks pengaruh total dimensi. Matriks pengaruh total dimensi ternormalisasi $T_D^\alpha = [t_{uv}^{\alpha D}]_{m \times m}$ dapat dihitung menggunakan rumus berikut.

$$t_{uv}^{\alpha D} = t_{uv}^D / \sum_{v=1}^m t_{uv}^D \quad (4.28)$$

Elemen $t_{uv}^{\alpha D}$ merupakan hasil pembagian elemen t_{uv}^D dengan penjumlahan baris elemen-elemen dalam matriks pengaruh total dimensi.

2. Pembentukan Supermatriks Tanpa Bobot

Kemudian, dilakukan pembentukan supermatriks tanpa bobot atau *unweighted supermatrix*. Supermatriks tanpa bobot $Q = [q_{ij}]_{n \times n}$ merupakan hasil transpose dari matriks pengaruh total kriteria ternormalisasi T_C^α .

$$Q = (T_C^\alpha)' \quad (4.29)$$

Sehingga, diperoleh supermatriks sebagai berikut.

$$Q = \begin{matrix} & \begin{matrix} D_1 & \dots & D_v & \dots & D_m \end{matrix} \\ \begin{matrix} D_1 \\ \vdots \\ D_u \\ D_m \end{matrix} & \begin{bmatrix} C_{11} & \dots & C_{1n'_1} & \dots & C_{v1} & \dots & C_{vn'_v} & \dots & C_{m1} & \dots & C_{mn} \\ C_{11} & \dots & C_{1n'_1} & \dots & C_{v1} & \dots & C_{vn'_v} & \dots & C_{m1} & \dots & C_{mn} \\ C_{1i} & \dots & C_{1n'_i} & \dots & C_{vi} & \dots & C_{vn'_v} & \dots & C_{mi} & \dots & C_{ni} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_{ui} & \dots & C_{un'_u} & \dots & C_{vi} & \dots & C_{vn'_v} & \dots & C_{mi} & \dots & C_{ni} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_{mi} & \dots & C_{mn} & \dots & C_{vi} & \dots & C_{vn'_v} & \dots & C_{mi} & \dots & C_{ni} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_{mn} & \dots & C_{mn} & \dots & C_{vi} & \dots & C_{vn'_v} & \dots & C_{mi} & \dots & C_{ni} \end{bmatrix} \end{matrix} \quad (4.30)$$

$$Q_C^{uv} = \begin{matrix} & \begin{matrix} C_{v1} & \dots & C_{vj} & \dots & C_{vn'_v} \end{matrix} \\ \begin{matrix} C_{u1} \\ \vdots \\ C_{ui} \\ \vdots \\ C_{un'_u} \end{matrix} & \begin{bmatrix} q_{11}^{uv} & \dots & q_{1j}^{uv} & \dots & q_{1n'_v}^{uv} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ q_{i1}^{uv} & \dots & q_{ij}^{uv} & \dots & q_{in'_v}^{uv} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{n'_u1}^{uv} & \dots & q_{n'_uj}^{uv} & \dots & q_{n'_un'_v}^{uv} \end{bmatrix} \end{matrix} \quad (4.31)$$

3. Pembentukan Supermatriks Berbobot

Supermatriks berbobot atau *weighted supermatrix* $Q^\alpha = [q_{ij}^\alpha]_{n \times n}$ merupakan hasil perkalian dari supermatriks tanpa bobot dan transpose matriks pengaruh total dimensi ternormalisasi $P = [p_{uv}]_{m \times m}$.

$$P = (T_D^\alpha)' \quad (4.32)$$

$$Q_C^{\alpha uv} = p_{uv} \times Q_C^{uv} \quad (4.33)$$

Sehingga, diperoleh supermatriks sebagai berikut.

$$Q^\alpha = \begin{matrix} & \begin{matrix} D_1 & \dots & D_v & \dots & D_m \end{matrix} \\ \begin{matrix} D_1 \\ \vdots \\ D_u \\ \vdots \\ D_m \end{matrix} & \begin{bmatrix} C_{11} & \dots & C_{1n'_1} & \dots & C_{v1} & \dots & C_{vn'_v} & \dots & C_{m1} & \dots & C_{mn} \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ C_{11} & \dots & C_{1i} & \dots & C_{1n'_1} & \dots & C_{u1} & \dots & C_{ui} & \dots & C_{un'_u} \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ C_{u1} & \dots & C_{ui} & \dots & C_{un'_u} & \dots & C_{m1} & \dots & C_{mi} & \dots & C_{mn} \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ D_m & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ & & C_{mn} & & & & & & & & \end{bmatrix} \end{matrix} \quad (4.34)$$

$$Q_C^{\alpha uv} = \begin{matrix} \begin{matrix} C_{u1} \\ \vdots \\ C_{ui} \\ \vdots \\ C_{un'_u} \end{matrix} & \begin{bmatrix} C_{v1} & \dots & C_{vj} & \dots & C_{vn'_v} \\ q_{11}^{\alpha uv} & \dots & q_{1j}^{\alpha uv} & \dots & q_{1n'_v}^{\alpha uv} \\ \vdots & & \vdots & & \vdots \\ q_{i1}^{\alpha uv} & \dots & q_{ij}^{\alpha uv} & \dots & q_{in'_v}^{\alpha uv} \\ \vdots & & \vdots & & \vdots \\ q_{n'_u1}^{\alpha uv} & \dots & q_{n'_uj}^{\alpha uv} & \dots & q_{n'_un'_v}^{\alpha uv} \end{bmatrix} \end{matrix} \quad (4.35)$$

4. Menghitung Limit dari Supermatrix

Selanjutnya, dilakukan perhitungan limit dari supermatriks berbobot atau *limiting matrix* $Q^W = [q_{ij}^W]_{n \times n}$. Perhitungan limit dari supermatriks ini dilakukan dengan mengalikan supermatriks tersebut dengan dirinya sendiri sebanyak φ . φ merupakan bilangan yang cukup besar sehingga nilai setiap baris dalam supermatriks dapat konvergen dan didapatkan supermatriks stabil jangka panjang atau *long-term stable supermatrix*.

$$Q^W = \lim_{\varphi \rightarrow \infty} (Q^\alpha)^\varphi \quad (4.36)$$

5. Menghitung Bobot Lokal dan Global

Bobot lokal kriteria didapatkan dari perhitungan rata-rata elemen baris pada supermatriks tanpa bobot Q.

$$w_i^C = \frac{1}{n} \sum_{j=1}^n q_{ij}^{\alpha} \quad (4.37)$$

Bobot lokal dimensi didapatkan dari perhitungan rata-rata elemen baris pada transpose matriks pengaruh total dimensi ternormalisasi P.

$$w_u^D = \frac{1}{m} \sum_{v=1}^m p_{uv} \quad (4.38)$$

Bobot global kriteria didapatkan dari rata-rata pada baris elemen konvergen dalam *limiting matrix* Q^W .

$$w_i = \frac{1}{n} \sum_{j=1}^n q_{ij}^W \quad (4.39)$$

C. Taguchi Loss Function

Tahapan fungsi kerugian Taguchi atau *Taguchi loss function* terdiri dari proses pengumpulan data performansi pemasok sampai pada mendapatkan *rating* setiap pemasok. Tahapan pengolahan metode *Taguchi loss function* pada kerangka kerja ini didasarkan pada penelitian Goren (2018).

1. Penilaian Kinerja Pemasok

Kinerja pemasok dapat dinilai berdasarkan pengalaman kerjasama dengan pemasok maupun berdasarkan pendapat ahli serta pihak pemangku kepentingan dalam perusahaan yang dikumpulkan melalui kuesioner atau bentuk pengumpulan data lainnya. Kinerja pemasok dinilai dalam bentuk persentase dengan rentang 0-100%. Hasil penilaian kinerja ini selanjutnya dirangkum dalam matriks keputusan atau *decision matrix* $D_{gi} = [d_{gi}]_{h \times n}$ di mana $g \in 1, 2, 3, \dots, h$ menandakan urutan pemasok dan $i \in 1, 2, 3, \dots, n$ menandakan urutan kriteria.

2. Penentuan *Loss Coefficient*

Sebelum menentukan koefisien kerugian atau *loss coefficient*, kriteria harus dikelompokkan dalam kategori resiko dan kategori manfaat. Hal ini dilakukan sehingga pembuat keputusan dapat menentukan nilai yang diinginkan, rentang nilai yang dapat diterima, serta batasan spesifikasi.

Untuk kriteria dalam kategori resiko, koefisien kerugian dapat dihitung dengan menggunakan rumus berikut.

$$k_i = \frac{L(Y_i)}{Y_i^2} \quad (4.40)$$

Di mana Y_i merupakan batasan spesifikasi atas atau *upper specification limit* yang ditentukan oleh pembuat keputusan dan $L(Y_i)$ merupakan nilai kerugian Taguchi pada batasan spesifikasi tersebut.

Untuk kriteria dalam kategori manfaat, koefisien kerugian dapat dihitung dengan menggunakan rumus berikut.

$$k_i = X_i^2 L(X_i) \quad (4.41)$$

Di mana X_i merupakan batasan spesifikasi bawah atau *lower specification limit* yang ditentukan oleh pembuat keputusan dan $L(X_i)$ merupakan nilai kerugian Taguchi pada batasan spesifikasi tersebut.

3. Perhitungan Skor Kerugian

Skor kerugian Taguchi individual dapat dihitung dengan menggunakan rumus berikut.

$$L(d_{gi}) = d_{gi} \times k_i^2 \quad (4.42)$$

Di mana d_{gi} merupakan nilai kinerja pemasok g terhadap kriteria i dan k_i merupakan koefisien kerugian untuk kriteria i .

Selanjutnya, dilakukan perhitungan skor kerugian Taguchi total. Hal ini dapat dilakukan dengan menggunakan rumus berikut.

$$TL_g = \sum_{i=1}^n L(d_{gi}) \times w_i \quad (4.43)$$

Di mana w_i merupakan bobot untuk kriteria i .

4. Penentuan *Rating* Pemasok

Rating pemasok dapat ditentukan dengan menggunakan rumus berikut.

$$W_g = \left(\frac{1}{TL_g} \right) / \sum_{g=1}^h \frac{1}{TL_g} \quad (4.44)$$

Nilai W_g selanjutnya digunakan dalam tahap pengolahan MO-MILP sebagai parameter nilai *rating* untuk pemasok ke- i yang disimbolkan dengan W_i .

4.2.4 Komponen Model Optimasi

Komponen dari model optimasi MO-MILP yang dikembangkan terdiri dari indeks, parameter, variabel keputusan, dan ukuran performansi. Ukuran performansi dari model pemilihan pemasok dan alokasi pesanan pada penelitian ini adalah total biaya gabungan atau *Joint Total Cost (JTC)* antara pamanufaktur dan pemasok, total nilai pembelian atau *Total Value Purchasing (TVP)*, dan jarak segregasi geografis atau *Geographical Segregation (GS)* sebagai salah satu tindakan preventif terhadap resiko disrupsi. Tabel berikut menjelaskan deskripsi dari setiap notasi yang digunakan dalam model optimasi yang dikembangkan. Notasi ini terdiri dari indeks, parameter, dan variabel keputusan.

Tabel 4.3 Notasi dalam Model Optimasi

Notasi	Deskripsi
1. Indeks	
I	Himpunan pemasok (<i>supplier</i>), $I = \{1,2,3, \dots, I \}$
R	Himpunan produk, $R = \{1,2,3, \dots, R \}$
T	Himpunan periode, $T = \{1,2,3, \dots, T \}$
S	Himpunan skenario disrupsi, $S = \{1,2,3, \dots, S \}$
2. Parameter	
α	Proporsi waktu disrupsi
ε	Faktor pembobotan (<i>weighting factors</i>) pada biaya terkait dengan pamanufaktur
δ	Faktor pembobotan (<i>weighting factors</i>) pada biaya terkait dengan pemasok
LL	Tingkat kehilangan penjualan (<i>lost sales</i>) maksimum
SD	Jarak segregasi terendah (<i>smallest segregation distance</i>) antara setiap pasangan pemasok (km)

CU	Biaya transportasi untuk setiap unit barang yang dikirim (rupiah/unit produk)
CD	Biaya bahan bakar untuk jarak tempuh yang dilalui (rupiah/km)
v	Biaya penalti untuk memasok produk berkualitas rendah (rupiah/unit produk)
Δ	Nilai ambang batas dari biaya ketahanan (<i>resilience cost</i>) untuk menghadapi disrupsi (rupiah)
Ψ	Interval waktu skenario disrupsi yang telah ditentukan (bulan)
ξ	Proporsi biaya kontingensi dari biaya penalti ketahanan (<i>resilience penalty cost</i>)
φ	Proporsi biaya mitigasi dari biaya penalti ketahanan
RC	Jumlah maksimum pemasok yang dapat dipilih
RL	Jumlah minimum pemasok yang dapat dipilih
M	Bilangan skalar bernilai besar (<i>large constant</i>)
τ_i^s	Fraksi kapasitas pemasok i yang bertahan setelah skenario s terjadi
D_{rt}	Permintaan produk r pada periode t (unit produk, UP)
d_{ij}	Jarak terdekat antara lokasi pemasok i dan j (km)
p^s	Probabilitas dari skenario disrupsi s
W_i	Nilai <i>rating</i> untuk pemasok ke- i
E_i	Kapasitas produksi primer pada pemasok ke- i (unit kapasitas, UK)
C_i	Waktu tersedia untuk setiap unit kapasitas untuk pemasok ke- i (menit/unit kapasitas)
g_i	Jarak antara pemasok i dengan perusahaan (km)
μ_i	Biaya operasional per unit kapasitas untuk pemasok ke- i (rupiah/unit kapasitas)
λ_i	Laju disrupsi yang diharapkan pada pemasok i (unit disrupsi/bulan)
f_i	Biaya pengadaan untuk kapasitas tambahan pada pemasok i (rupiah/unit kapasitas)
B_i	Biaya pembangunan saluran koneksi antara pemasok i dan perusahaan (rupiah)
G_i	Investasi primer yang diperlukan untuk membangun kapasitas restoratif tambahan pada pemasok i (rupiah/unit kapasitas)
Φ_i	Biaya penalti disrupsi yang dikeluarkan oleh pemasok i (rupiah/unit disrupsi)
c_i	Sumber daya yang diperlukan untuk memulihkan satu unit kapasitas dari pemasok i
η_i	Biaya sumber daya untuk pemulihan kapasitas dari pemasok i (rupiah/unit kapasitas)
θ_i	Tingkat kecacatan yang diharapkan pada produk dari pemasok i
A_i	Jumlah kapasitas restoratif yang tersedia di pemasok i (unit kapasitas)

AT_i	Waktu administrasi pada pemasok ke- i (menit)
TT_i	Waktu pengiriman pada pemasok ke- i (menit/km)
H_r	Biaya penyimpanan per unit untuk produk r pada perusahaan (rupiah/unit produk)
LO_r	Biaya kehilangan penjualan (<i>lost sales</i>) untuk produk r (rupiah/unit produk)
BI_r	Persediaan awal untuk produk r (unit produk)
IC_r	Kapasitas gudang persediaan maksimum untuk produk r (unit produk)
Q_r	Persentase kecacatan maksimum yang dapat diterima untuk produk r
LT_r	Waktu tunggu (<i>lead time</i>) maksimum yang dapat diterima untuk produk r (menit)
K_{ir}	Harga per unit untuk produk r pada pemasok ke- i (rupiah/unit produk)
O_{ir}	Biaya pemesanan untuk produk r pada pemasok ke- i (rupiah)
ST_{ir}	Waktu <i>set-up</i> produksi untuk produk r pada pemasok ke- i (menit)
PT_{ir}	Waktu produksi per unit untuk produk r pada pemasok ke- i (menit/unit produk)
3. Variabel Keputusan	
X_{irt}^s	Kuantitas pesanan untuk produk r dari pemasok ke- i yang dilakukan pada periode ke- t dan skenario s (unit produk)
Y_{irt}^s	Variabel biner yang bernilai 1 jika dan hanya jika dilakukan pemesanan dari pemasok ke- i untuk produk r pada periode ke- t dan skenario s
Z_i	Variabel biner yang bernilai 1 jika dan hanya jika pemasok i dipilih oleh perusahaan
I_{rt}^s	Tingkat persediaan (<i>inventory level</i>) untuk produk r pada akhir periode t dan skenario s (unit produk)
L_{rt}^s	Banyaknya kehilangan penjualan untuk produk r pada akhir periode t dan skenario s (unit produk)
N_i	Jumlah kejadian disrupsi yang terjadi pada pemasok i dalam periode waktu yang ditentukan (unit disrupsi, UD)
U_i	Kapasitas restoratif tambahan yang diinvestasikan oleh pemasok i (unit kapasitas)
r_{ij}	Variabel biner yang bernilai 1 jika dan hanya jika pemasok i dan j sama-sama terpilih
QC_{it}^s	Kapasitas kumulatif dari pemasok i pada periode t dan skenario s (unit kapasitas)
k_{it}^s	Kapasitas pemasok i yang dipulihkan pada periode t dan skenario s (unit kapasitas)
Ω^s	Biaya penalti ketahanan (<i>resilience penalty cost</i>) dalam skenario s (rupiah)

w_i	Variabel <i>auxiliary</i> terkait dengan jumlah disrupti pada pemasok i (unit disrupti)
ω_i	Kapasitas produksi tambahan yang disediakan oleh pemasok i selain dari kapasitas awal (unit kapasitas)
p_{irt}^s	Proporsi permintaan pelanggan terhadap produk r yang dilayani oleh pemasok i pada periode t dan skenario s
q_{rt}^s	Proporsi permintaan pelanggan terhadap produk r yang tidak terpenuhi pada periode t dan skenario s

Dalam referensi model yang dipakai, terdapat perbedaan penggunaan variabel keputusan untuk definisi yang sama. Dalam model yang dikembangkan oleh Goren (2018), variabel keputusan kuantitas pesanan produk dan banyaknya kehilangan penjualan masing-masing direpresentasikan dengan X_{irt} dan L_{rt} yang memiliki nilai bilangan bulat atau *integer*. Sedangkan dalam model yang dikembangkan oleh Hosseini, dkk. (2019), variabel keputusan ini direpresentasikan dengan p_{irt}^s dan q_{rt}^s yang masing-masing didefinisikan sebagai proporsi produk yang terlayani dan proporsi produk yang tidak terlayani dari permintaan (D_{rt}), sehingga kuantitas pesan dan kehilangan penjualan akan memiliki nilai bilangan real.

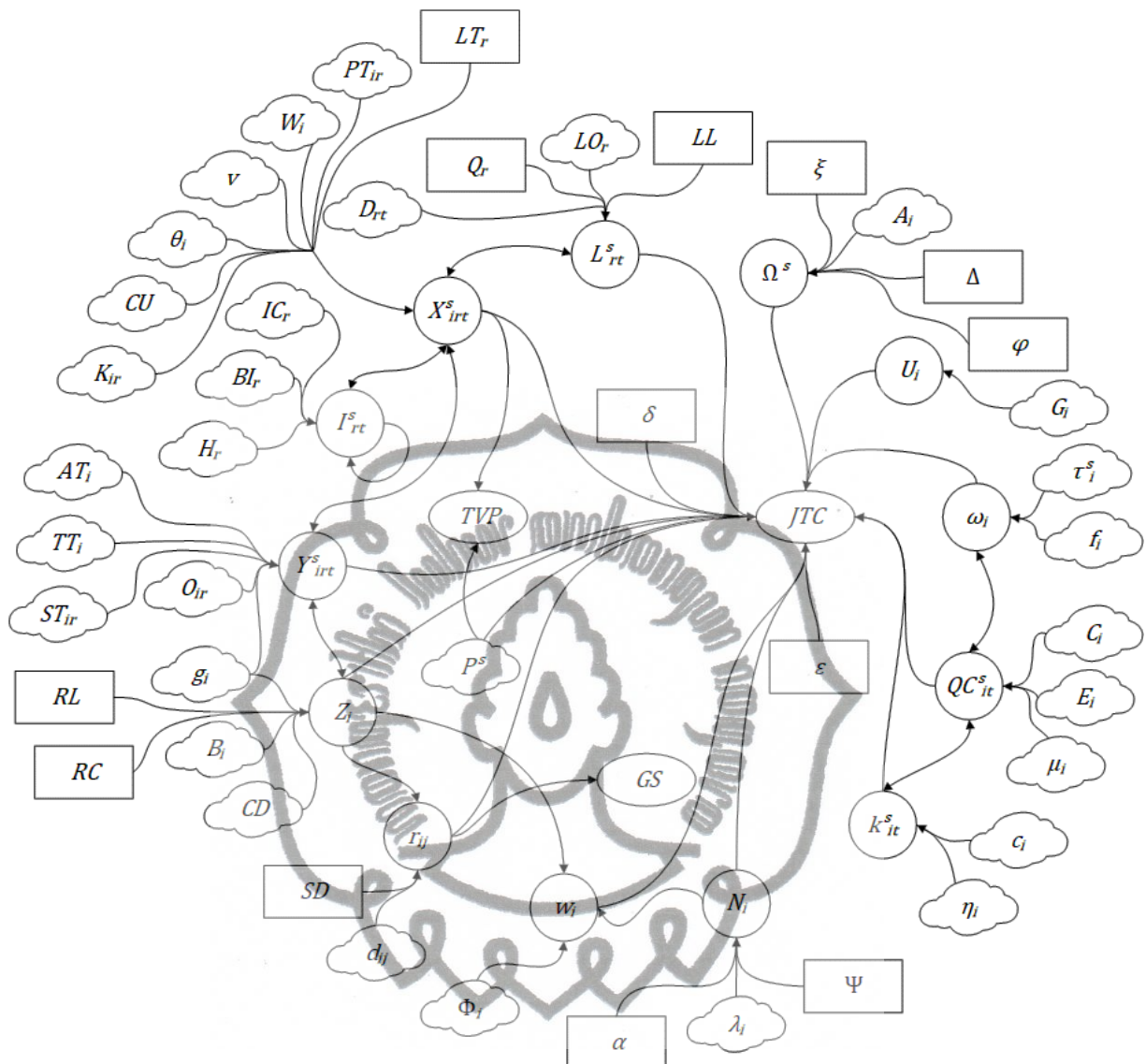
Model yang dikembangkan dalam penelitian ini mempertimbangkan tingkat persediaan pamanufaktur, karena itu akan lebih intuitif jika menggunakan nilai murni daripada proporsi. Selain itu, hasil solusi akan lebih baik bila disajikan dalam bentuk bilangan bulat, karena keputusan alokasi pesanan dinyatakan dalam satuan unit produk yang tidak memungkinkan untuk memiliki nilai desimal. Oleh karena itu pada penelitian ini, variabel keputusan kuantitas pesanan produk dan banyaknya kehilangan penjualan masing-masing direpresentasikan dengan X_{irt}^s dan L_{rt}^s dengan menambahkan indeks s untuk mewakili nilai pada setiap skenario disrupti.

$$X_{irt}^s = D_{rt} p_{irt}^s \quad (4.45)$$

$$L_{rt}^s = D_{rt} q_{rt}^s \quad (4.46)$$

4.2.5 Influence Diagram

Influence diagram merupakan sarana yang digunakan dapat membantu pemodel dalam eksplorasi kompleksitas dalam suatu sistem dan penyusunan model. *Influence diagram* menggambarkan karakteristik model dan hubungan antara *input* kendali, *input* tak terkendali, peubah acak, dan *output*.



Gambar 4.5 Influence Diagram

Input terkendali terdiri dari parameter tingkat kecacatan yang diperbolehkan, tingkat kehilangan penjualan yang diperbolehkan, jarak segregasi terkecil, jumlah minimum pemasok terpilih, jumlah maksimum pemasok terpilih, waktu tunggu maksimum yang diperbolehkan, proporsi waktu disrupsi, bobot biaya pamanufaktur, bobot biaya pemasok, interval waktu disrupsi, proporsi biaya mitigasi, proporsi biaya kontingensi, dan ambang batas dari biaya ketahanan. *Input* tak terkendali terdiri dari parameter model selain yang telah termasuk ke dalam *input* terkendali. Peubah acak terdiri dari variabel keputusan yang digunakan dalam model, hal ini termasuk variabel pemilihan pemasok, kuantitas pesanan pada setiap skenario, dan lain sebagainya. *Output* dari model berupa total biaya gabungan, total nilai pembelian, dan segregasi geografis.

4.2.6 Asumsi Model

Berikut adalah asumsi yang digunakan dalam model optimasi pemilihan pemasok dan alokasi pesanan:

1. Komponen biaya yang dipertimbangkan dalam model adalah: biaya pembelian barang, biaya penyimpanan barang, biaya pemesanan barang, biaya kapasitas tambahan, biaya penetapan koneksi, biaya penambahan kapasitas restoratif, biaya gangguan pada pemasok, biaya transportasi, biaya pemulihan pemasok, biaya operasional pemasok, biaya penalti ketahanan, biaya penalti produk cacat, dan biaya penalti kehilangan penjualan.
2. Bila terjadi skenario disrupsi ekstrem yang menghasilkan biaya ketahanan yang tinggi dan melampaui nilai ambang batas, maka akan dikenakan biaya penalti. Biaya penalti ini digunakan untuk menjalankan rencana mitigasi dan kontingensi untuk menghadapi disrupsi.
3. Biaya transportasi dihitung dari biaya transportasi per unit jarak dan biaya transportasi per unit produk.
4. Waktu tunggu terdiri dari komponen waktu administrasi, waktu *set-up*, waktu produksi, dan waktu pengiriman. Waktu administrasi terdiri dari waktu yang dihabiskan pada masa pesanan diterima sampai produksi dimulai serta waktu yang dihabiskan pada saat produk selesai diproduksi hingga mulai dikirim.
5. Model tidak mempertimbangkan *unexpected delay* pada saat proses administrasi, *set-up*, produksi, maupun transportasi.
6. Kapasitas produksi primer dan kapasitas produksi tambahan digunakan secara efektif sebelum dan saat terjadinya disrupsi, kecuali saat kapasitas ini gagal beroperasi dikarenakan disrupsi.
7. Kapasitas restoratif awal dan kapasitas restoratif tambahan adalah kapasitas yang dialokasikan saat fraksi kapasitas primer gagal berfungsi ketika terjadi disrupsi.

8. Kapasitas kumulatif merupakan rekaman jumlah kapasitas yang efektif beroperasi pada setiap tahap (*state*) disrupsi. Sedangkan, kapasitas yang dipulihkan (*restored capacity*) merupakan jumlah kapasitas produksi yang gagal beroperasi yang kemudian dipulihkan melalui ketersediaan kapasitas restoratif.
9. Untuk memulihkan satu unit kapasitas produksi dibutuhkan sumber daya pemulih yang dibayar dalam bentuk biaya penggerak sumber daya. Sumber daya yang digunakan untuk kegiatan pemulihan ini didefinisikan dalam bentuk kapasitas restoratif.
10. Kapasitas produksi pemasok dapat mengakomodasi seluruh jenis produk yang terdapat dalam himpunan R .
11. Nilai pembelian (*purchasing value*) dihitung berdasarkan banyaknya produk yang dipesan kepada pemasok dengan *rating* tertentu.
12. Fungsi tujuan segregasi geografis digunakan sebagai salah satu langkah preventif untuk memitigasi resiko disrupsi.
13. Disrupsi dapat terjadi pada setiap pemasok, waktu antar disrupsi yang dialami oleh pemasok diasumsikan mengikuti distribusi eksponensial.
14. Pada setiap periode, produk yang berbeda dari pemasok yang sama dikirim secara terpisah. Kebutuhan produk yang sama pada suatu periode dapat dipasok oleh beberapa pemasok yang berbeda.
15. Batasan atau kendala stokastik setidaknya mengandung proporsi waktu sebesar α .
16. Parameter biaya transportasi unit barang, biaya transportasi unit jarak, jumlah kapasitas primer, waktu kapasitas yang tersedia, biaya operasional, laju disrupsi, biaya pembangunan koneksi, investasi kapasitas restoratif, biaya disrupsi, sumber daya pemulihan, biaya sumber daya, tingkat kecacatan, kapasitas restoratif yang tersedia, waktu administrasi, waktu pengiriman, biaya penyimpanan, biaya kehilangan penjualan, persediaan awal, fraksi kapasitas yang bertahan, permintaan produk, harga unit produk, biaya pemesanan, waktu set-up, dan waktu produksi dianggap berdistribusi Uniform.

17. Strategi pemulihan akan secara optimal mengembalikan kapasitas pemasok yang terdisrupsi ke tingkat kapasitas awal.
18. Dalam menghadapi setiap skenario disrupsi dari waktu ke waktu, pemasok mengalami berbagai keadaan (*state*) yang diwakili oleh batasan evolusi keadaan pemasok. Keadaan ini mempengaruhi kapasitas kumulatif pemasok.
19. Parameter M memiliki nilai besar yang sekecil mungkin (*least big number*) tanpa melanggar konsistensi logika. Hal ini dilakukan untuk menghindari kesulitan komputasi (*numerical burden*) pada proses penyelesaian menggunakan perangkat lunak.
20. Batasan yang ketat atau batasan yang memiliki domain bertentangan dengan *range* fungsi tujuan akan memiliki *slack variable* dan setiap optimasi dilakukan terdapat fungsi tujuan tambahan yang bertujuan untuk meminimalkan *slack variable* tersebut.

4.2.7 Fungsi Tujuan

Dalam model yang dikembangkan terdapat tiga fungsi tujuan, yaitu: minimasi total biaya gabungan atau *Joint Total Cost (JTC)* antara pamanufaktur dan pemasok, maksimasi total nilai pembelian atau *Total Value Purchasing (TVP)*, dan maksimasi jarak segregasi geografis atau *Geographical Segregation (GS)*.

A. *Joint Total Cost*

Biaya total gabungan atau *joint total cost* terdiri dari biaya pembelian barang, biaya penyimpanan barang, biaya pemesanan barang, biaya kapasitas tambahan, biaya penetapan koneksi, biaya penambahan kapasitas restoratif, biaya gangguan pada pemasok, biaya transportasi, biaya pemulihan pemasok, biaya operasional pemasok, biaya penalti ketahanan, biaya penalti produk cacat, dan biaya penalti kehilangan penjualan.

1. Biaya Pembelian Barang

Total biaya pembelian barang atau *Purchasing Cost (PC)* merupakan hasil penjumlahan dari perkalian antara probabilitas skenario disrupsi tertentu (P^s), harga barang per unit untuk suatu produk yang dipesan melalui pemasok tertentu (K_{ir}), dan kuantitas

barang yang dipesan untuk produk tersebut melalui pemasok yang terpilih pada periode dan skenario gangguan tertentu (X_{irt}^s).

$$PC = \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} K_{ir} X_{irt}^s \right] \quad (4.47)$$

2. Biaya Penyimpanan Barang

Total biaya penyimpanan atau *Holding Cost (HC)* merupakan hasil penjumlahan dari perkalian antara probabilitas skenario gangguan tertentu (P^s), biaya penyimpanan per unit untuk suatu produk tertentu (H_r), dan tingkat persediaan untuk produk tersebut pada akhir periode dan skenario gangguan tertentu (I_{rt}^s).

$$HC = \sum_{s \in S} P^s \left[\sum_{r \in R} \sum_{t \in T} H_r I_{rt}^s \right] \quad (4.48)$$

3. Biaya Pemesanan Barang

Total biaya pemesanan atau *Ordering Cost (OC)* merupakan hasil penjumlahan dari perkalian antara probabilitas skenario gangguan tertentu (P^s), biaya pemesanan untuk suatu produk melalui pemasok tertentu (O_{ir}), dan keputusan berupa bilangan biner yang bernilai satu jika dan hanya jika dilakukan pemesanan melalui pemasok tersebut untuk produk terkait pada periode dan skenario gangguan tertentu (Y_{irt}^s), variabel ini merupakan variabel tanpa satuan atau *dimensionless*.

$$OC = \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} O_{ir} Y_{irt}^s \right] \quad (4.49)$$

4. Biaya Kapasitas Tambahan

Total biaya kapasitas tambahan pada pemasok atau *Additional Capacity Cost (ACC)* adalah biaya yang dikeluarkan pemasok untuk membangun unit kapasitas produksi tambahan. *ACC* merupakan hasil penjumlahan dari perkalian antara biaya pengadaan untuk kapasitas

tambahan pada pemasok tertentu (f_i) dan kapasitas produksi tambahan yang disediakan oleh pemasok tersebut selain dari kapasitas awal (ω_i).

$$ACC = \sum_{i \in I} f_i \omega_i \quad (4.50)$$

5. Biaya Penetapan Koneksi

Total biaya penetapan koneksi merupakan biaya yang dikeluarkan pemasok untuk menjalin hubungan dengan pemasok. Biaya untuk administrasi dan bea, perencanaan jalur distribusi, dan lain sebagainya dapat dipertimbangkan dalam biaya ini. Biaya penetapan koneksi atau *supplier connection cost* dilambangkan dengan SCC . SCC merupakan hasil penjumlahan dari perkalian antara biaya pembangunan koneksi antara pemasok tertentu dan perusahaan (B_i) serta pemasok yang dipilih oleh perusahaan (Z_i).

$$SCC = \sum_{i \in I} B_i Z_i \quad (4.51)$$

6. Biaya Penambahan Kapasitas Restoratif

Total biaya penambahan kapasitas restoratif atau *Restorative Capacity Investment (RCI)* merupakan jumlah investasi yang dikeluarkan pemasok untuk membangun unit kapasitas restoratif tambahan. RCI merupakan hasil penjumlahan dari perkalian antara investasi primer yang diperlukan untuk kapasitas restoratif pada pemasok tertentu (G_i) dengan kapasitas restoratif tambahan yang diinvestasikan oleh pemasok tersebut (U_i).

$$RCI = \sum_{i \in I} G_i U_i \quad (4.52)$$

7. Biaya Gangguan Pemasok

Total biaya gangguan yang dikeluarkan oleh pemasok atau *Disruption Cost (DC)* merupakan hasil penjumlahan dari perkalian antara biaya disrupsi yang dibebankan kepada pemasok tertentu (Φ_i)

karena, misal kegagalan memenuhi kontrak pesanan dan jumlah disrupsi yang terjadi pada pemasok tersebut dalam periode waktu yang ditentukan (N_i) serta keputusan berupa bilangan biner yang bernilai satu jika dan hanya jika pemasok tersebut dipilih oleh perusahaan (Z_i).

$$DC = \sum_{i \in I} Z_i N_i \Phi_i \quad (4.53)$$

Persamaan tersebut merupakan persamaan non-linear karena terdapat perkalian antara Z_i dan N_i yang merupakan variabel keputusan. Untuk mengatasi hal ini, didefinisikan sebuah variabel *auxiliary* yaitu w_i .

$$DC = \sum_{i \in I} w_i \Phi_i \quad (4.54)$$

Proposisi untuk komponen non-linear dalam persamaan terkait adalah bahwa komponen tersebut dapat dilinearisasi dengan menerapkan pernyataan $w_i = Z_i N_i$, yang berlaku dalam kendala berikut:

$$w_i \geq N_i - (1 - Z_i)M \quad \forall i \in I \quad (4.55)$$

Di mana M adalah konstanta dengan nilai yang besar, $Z_i \in \{0,1\}$, dan $N_i \in \mathbb{Z}$. Pembuktian dari proposisi di atas adalah sebagai berikut:

Kasus 1 adalah keadaan di mana $Z_i = 0$ dengan $\forall i \in I$. Pada kasus ini, didapati kendala $w_i \geq N_i - M$ atau mendekati $w_i \geq -M$ karena nilai M sangat besar, karena fungsi objektif terkait berupa minimasi maka $w_i = 0$. Kasus 2 adalah keadaan di mana $Z_i = 1$ dengan $\forall i \in I$. Pada kasus ini, didapati kendala $w_i \geq N_i$, karena fungsi objektif terkait berupa minimasi maka $w_i = N_i$. Maka terbukti bahwa $w_i = Z_i N_i$ berlaku di bawah kendala $w_i \geq N_i - (1 - Z_i)M$.

8. Biaya Transportasi

Total biaya transportasi atau *transportation cost* yang dilambangkan dengan TRC terdiri dari biaya pengiriman unit barang dan biaya pengiriman unit jarak tempuh. Biaya pengiriman per unit jarak tempuh merupakan hasil penjumlahan dari perkalian antara jarak

antara pemasok tertentu dengan perusahaan ($Z_i g_i$) dan biaya bahan bakar untuk jarak yang ditempuh (CD). Biaya pengiriman unit barang merupakan hasil penjumlahan dari perkalian jumlah barang yang dikirim (X_{irt}^s) dan biaya transportasi untuk setiap unit barang yang dikirim (CU).

$$TRC = \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} Z_i g_i CD + X_{irt}^s CU \right] \quad (4.56)$$

9. Biaya Pemulihan Pemasok

Total biaya pemulihan pemasok atau *Supplier Recovery Cost* (SRC) merupakan hasil penjumlahan dari perkalian antara probabilitas skenario disrupsi tertentu (P^s), biaya dan nilai sumber daya yang diperlukan untuk memulihkan satu unit kapasitas dari pemasok tertentu ($\eta_i c_i$), dan kapasitas pemasok terkait yang dipulihkan pada periode dan skenario tertentu (k_{it}^s).

$$SRC = \sum_{s \in S} P^s \left[\sum_{i \in I} \eta_i c_i \sum_{t \in T} k_{it}^s \right] \quad (4.57)$$

10. Biaya Operasional Pemasok

Total biaya operasional pemasok atau *operational cost* yang dilambangkan dengan SOC merupakan hasil penjumlahan dari perkalian antara probabilitas skenario disrupsi tertentu (P^s), biaya operasional per unit kapasitas pada pemasok tertentu (μ_i), dan jumlah kapasitas yang terpakai. Kapasitas yang terpakai merupakan penjumlahan kapasitas produksi primer (E_i) dengan kapasitas tambahan yang disediakan oleh pemasok tertentu (ω_i) dan dikurangi oleh kapasitas kumulatif dari pemasok terkait pada periode dan skenario tertentu (QC_{it}^s).

$$SOC = \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{t \in T} (E_i + \omega_i - QC_{it}^s) \mu_i \right] \quad (4.58)$$

11. Biaya Penalti Ketahanan

Total biaya penalti ketahanan atau *resilience penalty cost* merupakan biaya penalti pada skenario dengan biaya ketahanan *resilience cost* yang melebihi ambang batas maksimum (Δ). Biaya penalti ketahanan dihitung sebagai penjumlahan dari biaya mitigasi dan biaya kontingensi. biaya mitigasi merupakan biaya yang dikeluarkan sebelum terjadinya resiko untuk mengantisipasi resiko tersebut. Biaya mitigasi atau *Mitigation Cost* (MC) merupakan proporsi biaya mitigasi (φ) dari biaya penalti ketahanan pada suatu skenario (Ω^s).

$$MC = \sum_{s \in S} P^s \varphi \Omega^s \quad (4.59)$$

Sedangkan, biaya kontingensi merupakan biaya yang dikeluarkan saat terjadinya resiko. Biaya kontingensi atau *Contingency Cost* (CC) merupakan proporsi biaya kontingensi (ξ) dari biaya penalti ketahanan pada suatu skenario (Ω^s). Jumlah dari proporsi biaya mitigasi dan biaya kontingensi adalah satu.

$$CC = \sum_{s \in S} P^s \xi \Omega^s \quad (4.60)$$

12. Biaya Penalti Produk Cacat

Total biaya penalti produk cacat atau *defect penalty cost* yang dilambangkan dengan PDP merupakan hasil penjumlahan dari perkalian antara probabilitas skenario disrupsi tertentu (P^s), biaya penalti untuk memasok produk berkualitas rendah (v), dan jumlah produk cacat yang terkirim ($\theta_i X_{irt}^s$).

$$PDP = \sum_{s \in S} P^s \left[v \sum_{i \in I} \sum_{r \in R} \sum_{t \in T} \theta_i X_{irt}^s \right] \quad (4.61)$$

13. Biaya Penalti Kehilangan Penjualan

Pesanan kepada pemasok yang tidak terpenuhi akan menyebabkan kehilangan penjualan pada pihak pamanufaktur. Total biaya penalti atas kehilangan penjualan atau *Lost Sales Penalty* (LSP)

merupakan hasil penjumlahan dari perkalian antara probabilitas skenario disrupsi tertentu (P^s), biaya penalti kehilangan penjualan untuk suatu produk (LO_r), dan banyaknya kehilangan penjualan untuk produk tersebut pada akhir periode tertentu (L_{rt}^s).

$$LSP = \sum_{s \in S} P^s \left[\sum_{r \in R} \sum_{t \in T} LO_r L_{rt}^s \right] \quad (4.62)$$

Komponen biaya dalam total biaya yang telah dijelaskan dapat dikelompokkan menjadi biaya pamanufaktur dan biaya pemasok. Biaya pamanufaktur adalah biaya yang secara langsung dikeluarkan oleh pamanufaktur. Dalam persamaan total biaya, komponen biaya yang termasuk dalam biaya pamanufaktur adalah: biaya pembelian barang, biaya penyimpanan barang, biaya pemesanan barang, biaya penetapan koneksi, biaya transportasi, dan biaya penalti kehilangan penjualan. Berikut adalah persamaan untuk *Manufacturer Total Cost (MTC)*.

$$MTC = PC + HC + OC + SCC + TRC + LSP \quad (4.63)$$

$$\begin{aligned} MTC = & \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} K_{ir} X_{irt}^s \right] + \sum_{s \in S} P^s \left[\sum_{r \in R} \sum_{t \in T} H_r I_{rt}^s \right] \\ & + \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} O_{ir} Y_{irt}^s \right] + \sum_{i \in I} B_i Z_i \\ & + \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} Z_i g_i CD + X_{irt}^s CU \right] \\ & + \sum_{s \in S} P^s \left[\sum_{r \in R} \sum_{t \in T} LO_r L_{rt}^s \right] \end{aligned} \quad (4.64)$$

Sedangkan, biaya pemasok merupakan biaya yang secara langsung dikeluarkan oleh pemasok, dalam hal ini sebagian besar berkaitan dengan penalti yang harus dibayarkan kepada pamanufaktur. Komponen biaya yang merupakan biaya pemasok dalam total biaya adalah: biaya kapasitas tambahan, biaya investasi kapasitas restoratif, biaya gangguan pada pemasok,

biaya pemulihan pemasok, biaya operasional pemasok, dan biaya penalti produk cacat. Berikut adalah persamaan untuk *Supplier Total Cost (STC)*.

$$STC = ACC + RCI + DC + SRC + SOC + PDP \quad (4.65)$$

$$\begin{aligned} STC = & \sum_{i \in I} f_i \omega_i + \sum_{i \in I} G_i U_i + \sum_{i \in I} w_i \Phi_i + \sum_{s \in S} P^s \left[\sum_{i \in I} \eta_i c_i \sum_{t \in T} k_{it}^s \right] \\ & + \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{t \in T} (E_i + \omega_i - QC_{it}^s) \mu_i \right] \\ & + \sum_{s \in S} P^s \left[v \sum_{i \in I} \sum_{r \in R} \sum_{t \in T} \theta_i X_{irt}^s \right] \end{aligned} \quad (4.66)$$

Biaya penalti ketahanan tidak termasuk dalam pembebanan biaya pamanufaktur maupun pemasok, karena biaya penalti ketahanan sejatinya merupakan biaya yang dikeluarkan kedua belah pihak untuk mengantisipasi resiko disrupsi ekstrem (rencana mitigasi) dan menyelesaikan permasalahan yang terjadi saat disrupsi berlangsung (rencana kontingensi).

Karena adanya perbedaan dalam penanggung jawab biaya, didefinisikan sebuah parameter proporsi ε dan δ sebagai faktor pembobotan atau *weighting factors*. Masing-masing parameter tersebut merupakan faktor pembobotan untuk biaya terkait dengan pamanufaktur dan faktor pembobotan pada biaya terkait dengan pemasok. Dengan parameter tersebut, pemilik keputusan dapat menentukan sisi mana yang biayanya lebih diprioritaskan untuk dilakukan minimasi.

$$JTC = \varepsilon(MTC) + \delta(STC) + MC + CC \quad (4.67)$$

$$\begin{aligned} JTC = & \varepsilon \left(\sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} K_{ir} X_{irt}^s \right] + \sum_{s \in S} P^s \left[\sum_{r \in R} \sum_{t \in T} H_r I_{rt}^s \right] \right. \\ & + \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} Z_i g_i CD + X_{irt}^s CU \right] + \sum_{s \in S} P^s \left[\sum_{r \in R} \sum_{t \in T} LO_r L_{rt}^s \right] \\ & \left. + \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{r \in R} \sum_{t \in T} O_{ir} Y_{irt}^s \right] + \sum_{i \in I} B_i Z_i \right) + \dots \end{aligned}$$

$$\begin{aligned}
& \dots + \delta \left(\sum_{i \in I} f_i \omega_i + \sum_{i \in I} G_i U_i + \sum_{i \in I} w_i \Phi_i + \sum_{s \in S} P^s \left[\sum_{i \in I} \eta_i c_i \sum_{t \in T} k_{it}^s \right] \right. \\
& \quad + \sum_{s \in S} P^s \left[\sum_{i \in I} \sum_{t \in T} (E_i + \omega_i - Q C_{it}^s) \mu_i \right] \\
& \quad + \left. \sum_{s \in S} P^s \left[v \sum_{i \in I} \sum_{r \in R} \sum_{t \in T} \theta_i X_{irt}^s \right] \right) + \sum_{s \in S} P^s \varphi \Omega^s \\
& \quad + \sum_{s \in S} P^s \xi \Omega^s
\end{aligned} \tag{4.68}$$

Sehingga, fungsi objektif total biaya keseluruhan atau *Joint Total Cost (JTC)* dapat dituliskan dalam persamaan (4.69) sebagai berikut.

Minimize *JTC*

$$\begin{aligned}
& = \delta \left[\sum_{i \in I} f_i \omega_i + G_i U_i + w_i \Phi_i \right] + \varepsilon \left[\sum_{i \in I} B_i Z_i \right] \\
& \quad + \sum_{s \in S} \delta P^s \left[\sum_{i \in I} \sum_{t \in T} \left(\eta_i c_i k_{it}^s + (E_i + \omega_i - Q C_{it}^s) \mu_i + \sum_{r \in R} v \theta_i X_{irt}^s \right) \right] \\
& \quad + \sum_{s \in S} \varepsilon P^s \left[\sum_{r \in R} \sum_{t \in T} \left(\left(\sum_{i \in I} K_{ir} X_{irt}^s + O_{ir} Y_{irt}^s + (Z_i g_i CD + X_{irt}^s CU) \right) \right. \right. \\
& \quad \left. \left. + H_r I_{rt}^s + L O_r L_{rt}^s \right) \right] + \sum_{s \in S} P^s (\varphi + \xi) \Omega^s
\end{aligned} \tag{4.69}$$

B. *Total Value Purchasing*

Fungsi objektif *Total Value Purchasing (TVP)* bertujuan untuk memaksimalkan nilai pembelian total dalam horizon perencanaan. Dalam fungsi objektif ini, nilai *rating* pemasok (W_i) diperoleh dari fase awal pengolahan data melalui metode DEMATEL, DANP, dan *Taguchi Loss Function*. *TVP* mengukur total nilai pembelian atau *purchasing value* dari aspek keberlanjutan atau *three bottom line* yaitu aspek sosial, lingkungan, dan ekonomi dalam setiap unit barang yang dibeli melalui berbagai pemasok pada seluruh periode dan skenario disrupsi ($P^s X_{irt}^s$).

$$\text{Maximize } TVP = \sum_{i \in I} \sum_{r \in R} \sum_{t \in T} \sum_{s \in S} W_i P^s X_{irt}^s \quad (4.70)$$

C. *Geographical Segregation*

Penekanan resiko disrupsi dalam rantai pasok akibat terjadinya kejadian distupsi seperti bencana alam, bencana akibat manusia, atau kejadian lainnya dapat dilakukan dengan strategi ketahanan berupa segregasi geografis atau *Geographical Segregation (GS)*. *GS* seperti yang diperlihatkan pada persamaan (4.71) bertujuan untuk memaksimalkan total jarak (d_{ij}) antar pemasok yang terpilih (Z_i dan Z_j), sehingga bertindak untuk memisahkan posisi pemasok yang terpilih. Akibat dari lokasi antara pemasok yang terpilih tersebar secara geografis dengan jarak yang besar, bila suatu daerah terdampak oleh kejadian disrupsi, maka pemasok yang berada di daerah lain memiliki kemungkinan kecil terdampak oleh kejadian yang sama.

$$GS = \max_{i,j \leq a} \left(\sum_{i=1}^a \sum_{j=i+1}^a Z_i Z_j d_{ij} \right), \quad a = |I| \quad (4.71)$$

Bila fungsi objektif didasari pada persamaan tersebut, fungsi objektif menjadi non-linier karena terdapat perkalian antar variabel keputusan biner yaitu Z_i dan Z_j . Oleh karena itu harus dilakukan linearisasi terhadap persamaan tersebut, hal ini dapat dilakukan dengan memperkenalkan variabel biner *auxiliary* yaitu r_{ij} .

Proposisi untuk komponen non-linear dalam persamaan terkait adalah bahwa komponen tersebut dapat dilinearisasi dengan menerapkan pernyataan $r_{ij} = Z_i Z_j$ yang berlaku dalam himpunan kendala berikut:

$$r_{ij} \leq Z_i \quad \forall i, j \in I \mid i < j \quad (4.72)$$

$$r_{ij} \leq Z_j \quad \forall i, j \in I \mid i < j \quad (4.73)$$

$$r_{ij} \geq Z_i + Z_j - 1 \quad \forall i, j \in I \mid i < j \quad (4.74)$$

Di mana Z_i dan Z_j merupakan elemen dari himpunan bilangan biner atau $Z_i, Z_j \in \{0,1\}$. Pembuktian dari proposisi di atas adalah sebagai berikut:

Kasus 1 adalah keadaan di mana $Z_i = Z_j = 0$ dengan $\forall i, j \in I \mid i < j$. Pada kasus ini, terdapat kendala $r_{ij} \leq 0$ dan $r_{ij} \geq -1$ sehingga menghasilkan

nilai $r_{ij} = 0$. Kasus 2 adalah keadaan di mana $Z_i = 1$ dan $Z_j = 0$ atau $Z_i = 0$ dan $Z_j = 1$ dengan $\forall i, j \in I \mid i < j$. Pada kasus ini, terdapat kendala $r_{ij} \leq 0$ dan $r_{ij} \geq 0$ sehingga menghasilkan nilai $r_{ij} = 0$. Kasus 3 adalah keadaan di mana $Z_i = Z_j = 1$ dengan $\forall i, j \in I \mid i < j$. Pada kasus ini, terdapat kendala $r_{ij} \leq 1$ dan $r_{ij} \geq 1$ sehingga menghasilkan nilai $r_{ij} = 1$. Maka terbukti bahwa $r_{ij} = Z_i Z_j$ berlaku di bawah himpunan kendala $r_{ij} \leq Z_i$, $r_{ij} \leq Z_j$, dan $r_{ij} \geq Z_i + Z_j - 1$.

Sehingga fungsi objektif untuk memaksimalkan segregasi geografis dalam pemilihan pemasok dirumuskan pada persamaan (4.75) di bawah.

$$\text{Maximize } GS = \sum_{i \in I} \sum_{j \in I} r_{ij} d_{ij} \quad (4.75)$$

4.2.8 Himpunan Batasan

Dalam model yang dikembangkan terdapat beberapa himpunan batasan, meliputi: batasan persamaan kesetimbangan persediaan, batasan penetapan koneksi, batasan kualitas produk, kendala probabilitas disrupsi, batasan evolusi disrupsi, batasan kapasitas produksi, batasan tingkat pelayanan, batasan konsistensi nilai, dan batasan non-negatif.

A. Batasan Jarak Segregasi Terkecil

Parameter SD mewakili jarak segregasi terkecil antar lokasi dari setiap pasangan pemasok yang ditetapkan oleh pemilik keputusan. Batasan dalam persamaan (4.76) dirumuskan berdasarkan lokasi masing-masing pasangan pemasok potensial dengan menempatkan batas atas pada $SD = d_{ij}$ jika dan hanya jika kedua pemasok i dan j terpilih. Karena jika i dan j sama sama terpilih maka $(1 - Z_i)$ dan $(1 - Z_j)$ akan memiliki nilai nol. Namun, jika pemasok i atau j tidak terpilih, maka $(1 - Z_i)$, $(1 - Z_j)$, atau keduanya akan memiliki nilai 1. Dengan demikian batas atas pada SD dalam persamaan (4.76) akan memiliki nilai yang besar, yaitu sama dengan $d_{ij}(1 + M)$ atau $d_{ij}(1 + 2M)$, di mana M adalah konstanta dengan nilai yang besar. Oleh karena itu, hanya jarak antara dua pasang pemasok yang sama sama terpilih

yang akan memiliki efek pembatas pada parameter SD . Jumlah kendala yang terkait dengan persamaan (4.76) adalah $a(a-1)/2$ untuk $a = |I|$.

$$SD \leq d_{ij} (1 + M(1 - Z_i) + M(1 - Z_j)) \quad \forall i, j \in I \mid i < j \quad (4.76)$$

B. Batasan Persamaan Kesetimbangan Persediaan

Persamaan (4.77) dan (4.78) merupakan *inventory balance equation* (persamaan keseimbangan penyimpanan) yang menyatakan bahwa permintaan (D_{rt}) pada setiap periode dipenuhi melalui kuantitas barang yang dipesan (X_{irt}^s) atau melalui *on hand inventory* yang tersedia ($BI_r - I_{rt}^s$ untuk $t = 1$ dan $I_{rt-1}^s - I_{rt}^s$ untuk $t > 1$). Jika permintaan tidak dapat dipenuhi oleh pemasok maupun dari *on hand inventory*, batasan ini memastikan bahwa permintaan tersebut tercatat dalam jumlah penjualan yang hilang (L_{rt}^s).

$$BI_r - I_{rt}^s + L_{rt}^s + \sum_{i \in I} X_{irt}^s = D_{rt} \quad \forall r \in R, t = 1, \forall s \in S \quad (4.77)$$

$$I_{rt-1}^s - I_{rt}^s + L_{rt}^s + \sum_{i \in I} X_{irt}^s = D_{rt} \quad \forall r \in R, \forall t \in T/\{1\}, \forall s \in S \quad (4.78)$$

Pertidaksamaan berikut menyatakan bahwa tingkat persediaan produk tidak boleh melebihi kapasitas maksimal gudang persediaan.

$$I_{rt}^s \leq IC_r \quad \forall r \in R, \forall t \in T, \forall s \in S \quad (4.79)$$

C. Batasan Penetapan Koneksi

Batasan (4.80) menyatakan bahwa koneksi pada pemasok paling tidak sebanyak RL . Hal ini dilakukan untuk memastikan bahwa jika satu pemasok mengalami kegagalan dalam suatu skenario disrupsi, terdapat pemasok lainnya yang dapat menyediakan produk. Sementara batasan (4.81) menetapkan bahwa perusahaan hanya dapat terlibat dengan pemasok sebanyak RC .

$$\sum_{i \in I} Z_i \geq RL \quad (4.80)$$

$$\sum_{i \in I} Z_i \leq RC \quad (4.81)$$

D. Batasan Kualitas Produk

Batasan kualitas produk pada persamaan (4.82) memastikan bahwa rasio kecacatan (θ_i) dari produk yang dipesan dari setiap pemasok pada setiap periode (X_{irt}^s) tidak melebihi tingkat tertentu yang telah ditentukan oleh perusahaan seperti besar presentase kecacatan maksimum yang dapat diterima (Q_r) dari keseluruhan permintaan produk (D_{rt}).

$$\sum_{i \in I} X_{irt}^s \theta_i \leq Q_r D_{rt} \quad \forall r \in R, \forall t \in T, \forall s \in S \quad (4.82)$$

E. Kendala Probabilitas Disrupsi

Misalkan y_i adalah jumlah disrupsi yang terjadi pada pemasok i dalam interval waktu tertentu (Ψ), disrupsi ini mencakup gangguan eksternal maupun internal. Waktu antara disrupsi yang dialami oleh pemasok diasumsikan mengikuti distribusi eksponensial, di mana λ_i menunjukkan tingkat disrupsi yang diharapkan untuk pemasok i dalam interval waktu Ψ . Oleh karena itu, jumlah disrupsi untuk setiap pemasok mengikuti distribusi Poisson. Dengan demikian, probabilitas terjadinya disrupsi sebanyak N_i untuk pemasok i dengan interval waktu Ψ dihitung dengan persamaan berikut.

$$P(y_i = N_i) = f(N_i; \lambda_i \Psi) = \frac{(\lambda_i \Psi)^{N_i} e^{-\lambda_i \Psi}}{N_i!} \quad (4.83)$$

Karena sifat stokastik dari disrupsi, pendekatan *Chance Constraint Programming* (CCP) dapat digunakan. Asumsi utama dalam model CCP adalah bahwa batasan atau kendala stokastik setidaknya mengandung proporsi waktu sebesar α , di mana α adalah *safety margin* yang disesuaikan oleh pembuat keputusan. Berdasarkan konsep CCP, variabel stokastik y_i diganti dengan N_i sebagai variabel deterministik baru dan batasan kemungkinan (*chance constraint*) untuk memastikan bahwa jumlah disrupsi yang dialami pemasok i tidak akan melebihi N_i setidaknya dalam proporsi waktu sebesar α .

$$P(y_i \leq N_i) \geq \alpha \quad (4.84)$$

Sehingga didapatkan,

$$\sum_{k=0}^{N_i} \frac{(\lambda_i \Psi)^k e^{-\lambda_i \Psi}}{k!} \geq \alpha \quad (4.85)$$

atau

$$F(N_i; \lambda_i \Psi) \geq \alpha \quad (4.86)$$

Persamaan tersebut merupakan persamaan non-linear karena terdapat perkalian antara $1/N_i$ dan $(\lambda_i \Psi)^{N_i}$, sedangkan N_i sendiri merupakan variabel keputusan. Untuk mengatasi hal ini, dilakukan transformasi invers terhadap fungsi distribusi kumulatif Poisson dengan rata-rata sebesar $\lambda_i \Psi$ dan banyak kejadian sebesar N_i . Karena dilakukan transformasi invers maka variable N_i akan terisolasi di sisi kiri persamaan dan tidak akan terjadi perkalian antar kedua variable tersebut.

$$N_i \geq F^{-1}(\alpha; \lambda_i \Psi) \quad \forall i \in I \quad (4.87)$$

F. Batasan Evolusi Disrupsi

Kendala (4.88-4.90) mewakili evolusi keadaan pemasok dari waktu ke waktu dalam menghadapi setiap skenario disrupsi.

Kendala (4.88) menyatakan bahwa pada setiap periode selama skenario disrupsi berlangsung, baik kapasitas produksi primer (E_i) maupun kapasitas produksi tambahan (ω_i) akan memiliki penurunan kemampuan operasional akibat pengaruh dari disrupsi yang diwakili dengan fraksi kapasitas yang tersedia pasca disrupsi (τ_i^S). Sehingga kapasitas kumulatif yang dimiliki oleh pemasok (QC_{it}^S) adalah kapasitas yang tersedia pasca disrupsi ($\tau_i^S(E_i + \omega_i)$) dan kapasitas yang dipulihkan ($k_{i\sigma}^S$).

$$QC_{it}^S = \tau_i^S(E_i + \omega_i) + \sum_{1 \leq \sigma \leq t-1} k_{i\sigma}^S \quad \forall i \in I, \forall t \in T, \forall s \in S \quad (4.88)$$

Kendala (4.89) menyatakan bahwa pada setiap periode selama skenario disrupsi berlangsung, nilai gabungan antara kapasitas primer (E_i) dan kapasitas tambahan (ω_i) akan memiliki nilai yang lebih besar daripada kapasitas kumulatif (QC_{it}^S) karena adanya penurunan kemampuan operasional akibat disrupsi.

$$QC_{it}^S \leq E_i + \omega_i \quad \forall i \in I, \forall t \in T/\{|T| - 1\}, \forall s \in S \quad (4.89)$$

Kendala (4.90) menyatakan bahwa pada akhir periode skenario disrupsi, nilai gabungan antara kapasitas primer (E_i) dan kapasitas tambahan (ω_i) setara dengan kapasitas kumulatif (QC_{it}^s) akibat berakhirnya disrupsi.

$$QC_{it}^s = E_i + \omega_i \quad \forall i \in I, t = |T|, \forall s \in S \quad (4.90)$$

Batasan (4.91) digunakan untuk menentukan tingkat kapasitas restoratif dalam bentuk unit kapasitas setiap periode untuk masing-masing pemasok. Kapasitas yang dipulihkan (k_{it}^s) membutuhkan sebanyak c_i sumber daya yang berupa kapasitas restoratif (A_i) dan kapasitas restoratif tambahan (U_i).

$$c_i k_{it}^s \leq A_i + U_i \quad \forall i \in I, \forall t \in T, \forall s \in S \quad (4.91)$$

G. Batasan Penalti Ketahanan

Batasan (4.92) memastikan bahwa bila terjadi skenario disrupsi ekstrem yang menghasilkan biaya ketahanan (*resilience cost*) yang tinggi sehingga melampaui nilai ambang batas (Δ), maka akan dikenakan biaya penalti. *Resilience cost* ini terdiri dari biaya gangguan operasional bagi pamanufaktur (*operational disruption cost*) dan biaya pemulihan (*restoration cost*). *Resilience cost* berbeda dengan biaya penalti ketahanan (*resilience penalty cost*) yang dilambangkan dengan Ω^s . *Operational disruption cost* berupa biaya penalti atas kehilangan penjualan ($LO_r L_{rt}^s$) di pihak pamanufaktur karena pemasok tidak dapat mengirimkan produk yang telah dipesan serta biaya pembatalan pengiriman ($Z_i g_i CD + X_{irt}^s CU$). *Restoration cost* ($\eta_i c_i k_{it}^s$) berupa biaya yang dikeluarkan pemasok untuk melakukan pemulihan dari kejadian disrupsi.

$$\sum_{t \in T} \left[\sum_{i \in I} \eta_i c_i k_{it}^s + \sum_{r \in R} LO_r L_{rt}^s + \sum_{i \in I} \sum_{r \in R} (Z_i g_i CD + X_{irt}^s CU) \right] - \Delta \leq \Omega^s \quad \forall s \in S \quad (4.92)$$

H. Batasan Kapasitas Produksi

Pertidaksamaan (4.93) merupakan batasan kapasitas produksi yang menyatakan bahwa total kuantitas produk yang dipesan dari suatu pemasok pada suatu periode tertentu tidak dapat melebihi kapasitas pemasok pada periode terkait. Kapasitas pemasok diekspresikan dalam satuan waktu, sehingga total waktu produksi ($X_{irt}^s PT_{ir}$) dan total waktu *setup* ($Y_{irt}^s ST_{ir}$)

tidak dapat lebih besar daripada waktu kerja tersedia yang dimiliki oleh pemasok ($C_i QC_{it}^s$). Dimisalkan bahwa setiap unit kapasitas pemasok dapat memproduksi berbagai jenis produk dalam himpunan R yang akan dipesan oleh pamanufaktur.

$$\sum_{r \in R} (X_{irt}^s PT_{ir} + Y_{irt}^s ST_{ir}) \leq C_i QC_{it}^s \quad \forall i \in I, \forall t \in T, \forall s \in S \quad (4.93)$$

I. Batasan Tingkat Pelayanan

Tingkat layanan atau *service level* (SL) yang diinginkan merupakan rasio antara total penjualan ($D_{rt} - L_{rt}^s$) dan total permintaan (D_{rt}).

$$SL = \frac{D_{rt} - L_{rt}^s}{D_{rt}} \quad (4.94)$$

Tingkat layanan dapat diekspresikan dalam tingkat kehilangan penjualan atau *lost sales level* (LL) maksimum yang dimungkinkan.

$$SL = 1 - LL \quad (4.95)$$

Karena tingkat layanan yang sebenarnya harus lebih besar daripada tingkat layanan yang telah ditentukan oleh perusahaan maka,

$$\frac{D_{rt} - L_{rt}^s}{D_{rt}} \geq SL \quad (4.96)$$

$$1 - \frac{L_{rt}^s}{D_{rt}} \geq 1 - LL \quad (4.97)$$

$$-\frac{L_{rt}^s}{D_{rt}} \geq -LL \quad (4.98)$$

Sehingga,

$$L_{rt}^s / D_{rt} \leq LL \quad \forall r \in R, \forall t \in T, \forall s \in S \quad (4.99)$$

Batasan (4.99) tersebut menyatakan bahwa total penjualan yang hilang tidak dapat melebihi tingkat yang telah ditentukan oleh perusahaan yang disebut tingkat kehilangan penjualan maksimum.

J. Batasan waktu tunggu

Pertidaksamaan berikut merupakan batasan *lead time* atau waktu tunggu yang menyatakan bahwa total waktu tunggu mulai dari barang dipesan pada suatu pemasok pada suatu periode tertentu tidak boleh melebihi jangka

waktu tunggu yang telah ditentukan pada periode terkait. Waktu tunggu ini terdiri dari waktu tunggu administrasi pesanan (AT_i), waktu *set-up* (ST_{ir}) waktu produksi (PT_{ir}), dan waktu pengiriman (TT_i). Waktu administrasi terdiri dari waktu yang dihabiskan pada masa pesanan diterima sampai produksi dimulai serta waktu yang dihabiskan pada saat produk selesai diproduksi hingga mulai dikirim.

$$Y_{irt}^S(AT_i + ST_{ir} + g_i TT_i) + X_{irt}^S PT_{ir} \leq LT_r \quad \forall i \in I, \forall r \in R, \forall t \in T, \forall s \in S \quad (4.100)$$

K. Batasan Konsistensi Nilai dan Batasan Non-negatif

Batasan (4.101) memastikan konsistensi logika (*logic consistency*) dari variabel X_{irt}^S dan Y_{irt}^S . Jika variabel Y_{irt}^S memiliki nilai 1 atau $Y_{irt}^S = 1$ maka variabel X_{irt}^S pasti memiliki nilai lebih dari atau sama dengan 0 atau $X_{irt}^S \geq 0$. Sedangkan, jika variabel Y_{irt}^S memiliki nilai 0 atau $Y_{irt}^S = 0$ maka variabel X_{irt}^S tidak mungkin memiliki nilai lebih dari 0 atau $X_{irt}^S = 0$. Hal ini berarti pemesanan sebesar X_{irt}^S unit untuk produk j dari pemasok ke- i dapat dilakukan pada periode ke- t dan skenario disrupsi s jika dan hanya jika pamanufaktur memilih pemasok ke- i untuk menyediakan produk j tersebut pada periode dan dan skenario disrupsi terkait.

$$X_{irt}^S \leq Y_{irt}^S M \quad \forall i \in I, \forall r \in R, \forall t \in T, \forall s \in S \quad (4.101)$$

Batasan (4.102) memastikan konsistensi logika dari variabel Z_i dan Y_{irt}^S . Jika variabel Z_i memiliki nilai 1 atau $Z_i = 1$ maka variabel Y_{irt}^S pasti memiliki nilai lebih dari atau sama dengan 0, karena Y_{irt}^S merupakan variable biner maka nilai yang sesuai adalah antara 0 atau 1 sehingga $Y_{irt}^S = \{0,1\}$. Sedangkan, jika variabel Z_i memiliki nilai 0 atau $Z_i = 0$ maka variabel Y_{irt}^S tidak mungkin memiliki nilai lebih dari 0 atau $Y_{irt}^S = 0$. Hal ini berarti variabel Y_{irt}^S dapat memiliki nilai 1 jika dan hanya jika variabel Z_i juga memiliki nilai 1, tetapi bila variabel Z_i memiliki nilai 1, variabel Y_{irt}^S belum tentu memiliki nilai 1 juga.

$$Y_{irt}^S \leq Z_i \quad \forall i \in I, \forall r \in R, \forall t \in T, \forall s \in S \quad (4.102)$$

Selain itu, terdapat pula batasan untuk konsistensi nilai pada *auxiliary variable* w_i yang dirumuskan pada persamaan (4.55) serta r_{ij} yang

dirumuskan pada persamaan (4.72-4.74). Proposisi kedua himpunan Batasan tersebut telah dibuktikan dalam bagian penjelasan fungsi tujuan.

Batasan (4.103-4.104) merupakan batasan non-negatif yang memastikan bahwa variabel keputusan $X_{irt}^s, I_{rt}^s, L_{rt}^s, \omega_i, N_i, U_i, w_i, QC_{it}^s, \Omega^s$, dan k_{it}^s memiliki nilai nol atau positif.

$$X_{irt}^s, I_{rt}^s, L_{rt}^s, \omega_i, N_i, U_i, w_i \geq 0 \quad \forall i \in I, \forall r \in R, \forall t \in T, \forall s \in S \quad (4.103)$$

$$QC_{it}^s, \Omega^s, k_{it}^s \geq 0 \quad \forall i \in I, \forall t \in T, \forall s \in S \quad (4.104)$$

Batasan (4.105) memastikan bahwa variabel keputusan Y_{irt}^s, Z_i , dan r_{ij} memiliki nilai yang berupa bilangan biner (0 atau 1).

$$Y_{irt}^s, Z_i, r_{ij} \in \{0,1\} \quad \forall i \in I, \forall r \in R, \forall t \in T, \forall s \in S \quad (4.105)$$

Batasan (4.106-4.107) memastikan bahwa variabel keputusan $X_{irt}^s, I_{rt}^s, L_{rt}^s, \omega_i, N_i, U_i, w_i, QC_{it}^s$, dan k_{it}^s memiliki nilai yang berupa bilangan bulat (*integer*).

$$X_{irt}^s, I_{rt}^s, L_{rt}^s, \omega_i, N_i, U_i, w_i \in \mathbb{Z} \quad \forall i \in I, \forall r \in R, \forall t \in T, \forall s \in S \quad (4.106)$$

$$QC_{it}^s, k_{it}^s \in \mathbb{Z} \quad \forall i \in I, \forall t \in T, \forall s \in S \quad (4.107)$$

4.2.9 Prosedur Penyelesaian Optimasi

Berikut merupakan prosedur penyelesaian untuk mendapatkan solusi optimal dari fungsi tujuan majemuk.

1. Selesaikan optimasi model dengan hanya menggunakan fungsi objektif $\min(JTC)$, nilai optimum dari penyelesaian ini adalah JTC_{\min} .
2. Selesaikan optimasi model dengan hanya menggunakan fungsi objektif $\min(TVP)$ lalu $\max(TVP)$, nilai optimum dari penyelesaian ini adalah TVP_{\min} serta TVP_{\max} .
3. Selesaikan optimasi model dengan hanya menggunakan fungsi objektif $\min(GS)$ lalu $\max(GS)$, nilai optimum dari penyelesaian ini adalah GS_{\min} serta GS_{\max} .
4. Karena $\max(JTC)$ memiliki nilai solusi tanpa batas (*unbounded*), maka normalisasi fungsi tujuan JTC adalah sebagai berikut.

$$JTC' = \lim_{JTC_{\max} \rightarrow \infty} \left(\frac{JTC - JTC_{\min}}{JTC_{\max} - JTC_{\min}} \right) = \frac{JTC - JTC_{\min}}{JTC_{\max} - JTC_{\min}} \quad (4.108)$$

5. Sementara, normalisasi fungsi tujuan TVP dan GS adalah sebagai berikut.

$$TVP' = \frac{TVP_{\max} - TVP}{TVP_{\max} - TVP_{\min}} \quad (4.109)$$

$$GS' = \frac{GS_{\max} - GS}{GS_{\max} - GS_{\min}} \quad (4.110)$$

6. Terakhir, lakukan optimasi terhadap fungsi tujuan transformasi atau fungsi tujuan agregat, fungsi tujuan agregat didefinisikan sebagai berikut.

$$\text{Minimize } f' = JTC' + TVP' + GS' \quad (4.111)$$

4.3 VERIFICATION, VALIDATION, AND TESTING (VVT)

Bagian ini menjelaskan mengenai proses verifikasi (*verifikasi*), validasi (*validation*), dan pengujian (*testing*) model MO-MILP. Proses verifikasi dilakukan untuk mengetahui validitas internal model dan proses validasi dilakukan untuk mengetahui validitas eksternal model.

4.3.1 Preliminary Testing

Pengujian awal (*preliminary testing*) dilakukan untuk mengetahui apakah interpretasi model yang dibuat dalam bentuk kode pemrograman dapat berjalan dengan baik. Model MO-MILP diinterpretasikan dalam kode pemrograman Python 3 dengan menggunakan ekstensi Gurobi. Kode pemrograman yang telah dirancang terdapat pada Lampiran 1 Kode Pemrograman Model MO-MILP SS-OAP.

Interpretasi pemrograman yang telah dirumuskan menghasilkan kode yang dapat dijalankan, akan tetapi algoritma kesulitan dalam mendapatkan nilai *best bound*. Oleh karena itu, dilakukan relaksasi setiap batasan untuk mengetahui batasan mana yang membutuhkan *slack variable*. Relaksasi batasan dilakukan dengan menggunakan fungsi `feasRelaxS`, fungsi ini melakukan relaksasi dengan menambahkan variabel buatan (*artificial variable*) “`art_`” pada sisi kanan atau *Right Hand Side (RHS)* dan sisi kiri atau *Left Hand Side (LHS)* dari persamaan atau pertidaksamaan.

$$LHS + \text{art}(x_l) \leq RHS + \text{art}(x_r) \quad (4.112)$$

Konfigurasi parameter `feasRelaxS` yang digunakan adalah `(0, False, False, True)` yang memiliki arti: meminimasi ukuran pelanggaran batas variabel (*variable bound*) dan kendala (*constraint*), meminimasi biaya pelanggaran (*violation cost*), batas variabel tidak dapat direlaksasi, dan kendala dapat direlaksasi.

```
pr1 = gp.read("JTCmin.lp")
pr1.feasRelaxS(0, False, False, True)
pr1.optimize()
JTCminval = pr1.objVal
JTCbstbnd = pr1.objBound
```

Dari hasil eksekusi program, didapatkan bahwa batasan (4.88) merupakan batasan yang memerlukan *slack variable*. Hal ini dikarenakan pada *LHS*, variabel QC_{it}^s merupakan variabel bilangan bulat (*integer*), sedangkan *RHS* akan berupa bilangan real karena τ_i^s merupakan variabel kontinu. Sehingga, dibutuhkan *slack variable* σQ_{it}^s (*artQ*) yang merupakan variabel kontinu sebagai pembulat dari nilai yang dihasilkan pada *RHS*.

```
artQ = {(i,t,s): m.addVar(vtype=GRB.CONTINUOUS,
                        name="artQ_{0}_{1}_{2}".format(i,t,s),
                        lb=0)
        for i in I for t in T for s in S}
```

Sehingga, batasan (4.88) diubah menjadi:

$$QC_{it}^s = \tau_i^s(E_i + \omega_i) + \sigma Q_{it}^s + \sum_{1 \leq \sigma \leq t-1} k_{i\sigma}^s \quad \forall i \in I, \forall t \in T, \forall s \in S \quad (4.113)$$

Serta digunakan batasan tambahan berikut:

$$\sigma Q_{it}^s \leq 1 \quad \forall i \in I, \forall t \in T, \forall s \in S \quad (4.114)$$

```
QCxC = {}
ckxC = {}
for i in I:
    for t in T:
        for s in S:
            t1 = 1
            ck = {}
            ck[i,t,s] = 0
            while 1<=t1<=(t-1):
                ck[i,t,s] += k[i,t1,s]
                t1 += 1
            QCxC[i,t,s] = m.addConstr(QC[i,t,s]==ck[i,t,s]+
                                     tau[i,s]*(E[i]+omg[i])+artQ[i,t,s],
                                     "QCxC_{0}_{1}_{2}".format(i,t,s))
            aQxC = m.addConstr(artQ[i,t,s] <= 1,
                               "aQxC_{0}_{1}_{2}".format(i,t,s))
```

Batasan (4.114) ditambahkan agar nilai *slack variable* σQ_{it}^s kecil, di mana $\sigma Q_{it}^s \approx 0$ dan $\sigma Q_{it}^s < 1$. Sehingga, nilai σQ_{it}^s akan dianggap sebagai 1 unit kapasitas yang bertahan pada awal mula disrupsi akibat dari nilai kapasitas bertahan $\tau_i^s(E_i + \omega_i)$ yang dibulatkan keatas (*ceiling*) oleh σQ_{it}^s pada persamaan (4.113).

4.3.2 Validasi

Validasi dilakukan untuk memeriksa apakah model sudah mencukupi untuk mewakili sistem nyata dan apakah model memberikan informasi dan jawaban yang benar. Karena sistem nyata untuk model yang dikembangkan belum ada, maka validitas eksternal diperiksa melalui kemampuan model untuk dapat berjalan dengan baik sehingga dapat memberikan solusi yang optimal. Oleh karena itu, validasi eksternal dilakukan melalui pemeriksaan feasibilitas model dan memeriksa inkonsistensi dalam batasan.

A. *Feasibility Check*

Pemeriksaan feasibilitas (*feasibility check*) model bertujuan untuk mengetahui apakah model dapat diselesaikan oleh *solver* dengan kata lain feasibel. Pemeriksaan feasibilitas dapat dilakukan dengan menggunakan kode pemrograman berikut.

```
tr = gp.read("TRF.lp")
tr.optimize()
status = tr.status
if status in (GRB.INF_OR_UNBD, GRB.INFEASIBLE,
              GRB.UNBOUNDED):
    print("Status: Aborted")
    print("Message: %d" % status)
if status == GRB.OPTIMAL:
    print("Status: Optimal")
if status != (GRB.OPTIMAL or GRB.UNBOUNDED or
              GRB.INF_OR_UNBD or GRB.INFEASIBLE):
    print("Status : Stopped")
    print("Message: %d" % status)
print("Optimum Value: %g" % tr.objVal)
```

Dari hasil pemeriksaan feasibilitas model diketahui bahwa model memiliki status feasibel (*feasible*) yang berarti bahwa interpretasi model dalam kode pemrograman memiliki solusi yang memenuhi batasan.

B. *Inconsistent Subsystem*

Bila hasil pemeriksaan feasibilitas menyatakan bahwa model tidak feasibel (*infeasible*) maka dapat dilakukan pemeriksaan subsistem tidak dapat tereduksi yang tidak konsisten atau *Irreducible Inconsistent Subsystem* (IIS). IIS merupakan himpunan bagian (*subset*) kendala dan batas variabel yang memiliki dua sifat berikut:

1. Subsistem yang direpresentasikan dalam IIS tidak feasibel.
2. Jika kendala atau batas variabel dalam IIS dihilangkan, subsistem dapat menjadi feasibel.

Model yang tidak feasibel dapat memiliki beberapa IIS. Pemeriksaan IIS dapat dilakukan dengan menggunakan kode pemrograman berikut.

```
if status == (GRB.INFEASIBLE or GRB.INF_OR_UNBD):
    print("CHECKING IIS")
    tr.computeIIS()
    if tr.IISMinimal:
        print("IIS is minimal\n")
    else:
        print("IIS is not minimal\n")
    print("Unsatisfied/Breached constraints:")
    for cnst in tr.getConstrs():
        if cnst.IISConstr:
            print("%s" % cnst.constrName)
```

Dari hasil pemeriksaan IIS diketahui bahwa model tidak memiliki subsistem tidak dapat tereduksi yang tidak konsisten.

4.3.3 Verifikasi

Verifikasi dilakukan untuk memeriksa bahwa model benar secara logis dan matematis. Verifikasi dilakukan dengan memastikan konsistensi dimensi, memeriksa kesalahan pada penyusunan sintaks kode pemrograman, serta memeriksa interpretasi model dalam pemrograman secara manual.

A. Konsistensi Dimensi

Konsistensi dimensi dilakukan untuk memastikan bahwa dimensi atau satuan dari seluruh parameter atau variable dalam suatu persamaan atau pertidaksamaan tereduksi sehingga memiliki nilai yang sama pada sisi kanan dan sisi kiri dari persamaan atau pertidaksamaan tersebut. Dalam pembuktian [] memiliki arti bahwa parameter atau variabel bersangkutan tidak memiliki dimensi atau *dimensionless*. [UP] merupakan dimensi unit produk, [UK]

merupakan dimensi unit kapasitas, dan [UD] merupakan dimensi unit disrupsi. Tabel berikut menampilkan pembuktian konsistensi dimensi dari setiap persamaan atau pertidaksamaan dalam model MO-MILP.

Tabel 4.4 Pemeriksaan Konsistensi Dimensi

Persamaan	Pembuktian
(4.69)	$JTC[\text{rupiah}] = \delta[\] (f_i[\text{rupiah/UK}] \omega_i[\text{UK}] + G_i[\text{rupiah/UK}] U_i[\text{UK}]$ $+ w_i[\text{UD}] \phi_i[\text{rupiah/UD}]) + \varepsilon[\] (B_i[\text{rupiah}] Z_i[\])$ $+ \delta[\] P^s[\] (\eta_i[\text{rupiah/UK}] c_i[\] k_{it}^s[\text{UK}]$ $+ (E_i[\text{UK}] + \omega_i[\text{UK}] - QC_{it}^s[\text{UK}]) \mu_i[\text{rupiah/UK}]$ $+ v[\text{rupiah/UP}] \theta_i[\] X_{irt}^s[\text{UP}])$ $+ \varepsilon[\] P^s[\] (H_r[\text{rupiah/UP}] I_{rt}^s[\text{UP}]$ $+ LO_r[\text{rupiah/UP}] L_{rt}^s[\text{UP}]$ $+ K_{ir}[\text{rupiah/UP}] X_{irt}^s[\text{UP}] + O_{ir}[\text{rupiah}] Y_{irt}^s[\]$ $+ Z_i[\] g_i[\text{km}] CD[\text{rupiah/km}]$ $+ X_{irt}^s[\text{UP}] CU[\text{rupiah/UP}])$ $+ P^s[\] (\varphi[\] + \xi[\]) \Omega^s[\text{rupiah}]$ $[\text{rupiah}] = [\] ([\text{rupiah}] + [\text{rupiah}] + [\text{rupiah}]) + [\] ([\text{rupiah}])$ $+ [\] ([\text{rupiah}] + ([\text{UK}] + [\text{UK}] - [\text{UK}]) [\text{rupiah/UK}]$ $+ [\text{rupiah}])$ $+ [\] ([\text{rupiah}] + [\text{rupiah}] + [\text{rupiah}] + [\text{rupiah}]$ $+ [\text{rupiah}] + [\text{rupiah}]) + [\] ([\] + [\])[\text{rupiah}]$ $[\text{rupiah}] = [\] ([\text{rupiah}]) + [\] ([\text{rupiah}]) + [\] ([\text{rupiah}])$ $+ [\] ([\text{rupiah}]) + [\text{rupiah}]$ $[\text{rupiah}] = [\text{rupiah}] + [\text{rupiah}] + [\text{rupiah}] + [\text{rupiah}]$ $+ [\text{rupiah}]$ $[\text{rupiah}] = [\text{rupiah}] \quad \therefore \text{dimensi konsisten}$
(4.70)	$TVP[\text{UP}] = W_i[\] P^s[\] X_{irt}^s[\text{UP}]$ $[\text{UP}] = [\text{UP}] \quad \therefore \text{dimensi konsisten}$
(4.75)	$GS[\text{km}] = r_{ij}[\] d_{ij}[\text{km}]$ $[\text{km}] = [\text{km}] \quad \therefore \text{dimensi konsisten}$
(4.76)	$SD[\text{km}] \leq d_{ij}[\text{km}] \left(([\] + M[\])([\] - Z_i[\]) + M[\]([\] - Z_j[\]) \right)$ $[\text{km}] \leq [\text{km}]([\] + [\] + [\])$ $[\text{km}] \leq [\text{km}] \quad \therefore \text{dimensi konsisten}$

(4.77)	$D_{rt}[\text{UP}] = BI_r[\text{UP}] - I_{rt}^s[\text{UP}] + L_{rt}^s[\text{UP}] + X_{irt}^s[\text{UP}]$ $[\text{UP}] = [\text{UP}] \quad \therefore \text{dimensi konsisten}$
(4.78)	$D_{rt}[\text{UP}] = I_{rt-1}^s[\text{UP}] - I_{rt}^s[\text{UP}] + L_{rt}^s[\text{UP}] + X_{irt}^s[\text{UP}]$ $[\text{UP}] = [\text{UP}] \quad \therefore \text{dimensi konsisten}$
(4.79)	$I_{rt}^s[\text{UP}] \leq IC_r[\text{UP}] \quad \therefore \text{dimensi konsisten}$
(4.80)	$Z_i[\] \geq RL[\] \quad \therefore \text{dimensi konsisten}$
(4.81)	$Z_i[\] \leq RC[\] \quad \therefore \text{dimensi konsisten}$
(4.82)	$X_{irt}^s[\text{UP}] \theta_i[\] \leq Q_r[\] D_{rt}[\text{UP}]$ $[\text{UP}] \leq [\text{UP}] \quad \therefore \text{dimensi konsisten}$
(4.87)	$\sum_{k=0}^{N_i} \frac{(\lambda_i \Psi)^k e^{-\lambda_i \Psi}}{k!} \geq \alpha$ $\alpha[\] \leq \frac{(\lambda_i[\text{UD/bulan}] \Psi[\text{bulan}])^{N_i[\text{UD}]} e^{-\lambda_i[\text{UD/bulan}] \Psi[\text{bulan}]} }{N_i[\text{UD}]!}$ $[\] \leq \frac{([\text{UD}])^{[\text{UD}]} [\]^{[\text{UD}]} }{[\text{UD}]!}$ $[\] \leq [\] \quad \therefore \text{dimensi konsisten}$
(4.88)	$QC_{it}^s[\text{UK}] = k_{i\mu}^s[\text{UK}] + \tau_i^s[\] (E_i[\text{UK}] + \omega_i[\text{UK}])$ $[\text{UK}] = [\text{UK}] \quad \therefore \text{dimensi konsisten}$
(4.89)	$QC_{it}^s[\text{UK}] \leq E_i[\text{UK}] + \omega_i[\text{UK}]$ $[\text{UK}] \leq [\text{UK}] \quad \therefore \text{dimensi konsisten}$
(4.90)	$QC_{it}^s[\text{UK}] = E_i[\text{UK}] + \omega_i[\text{UK}]$ $[\text{UK}] = [\text{UK}] \quad \therefore \text{dimensi konsisten}$
(4.91)	$c_i[\] k_{it}^s[\text{UK}] \leq A_i[\text{UK}] + U_i[\text{UK}]$ $[\text{UK}] \leq [\text{UK}] \quad \therefore \text{dimensi konsisten}$
(4.92)	$\eta_i[\text{rupiah/UK}] c_i[\] k_{it}^s[\text{UK}] + LO_r[\text{rupiah/UP}] L_{rt}^s[\text{UP}]$ $+ Z_i[\] g_i[\text{km}] CD[\text{rupiah/km}]$ $+ X_{irt}^s[\text{UP}] CU[\text{rupiah/UP}] - \Delta[\text{rupiah}]$ $\leq \Omega^s[\text{rupiah}]$ $[\text{rupiah/UK}] [\text{UK}] + [\text{rupiah/UP}] [\text{UP}] + [\text{km}] [\text{rupiah/km}]$ $+ [\text{UP}] [\text{rupiah/UP}] - [\text{rupiah}] \leq [\text{rupiah}]$ $[\text{rupiah}] + [\text{rupiah}] + [\text{rupiah}] + [\text{rupiah}] - [\text{rupiah}] \leq [\text{rupiah}]$ $[\text{rupiah}] \leq [\text{rupiah}] \quad \therefore \text{dimensi konsisten}$

(4.93)	$X_{irt}^s[\text{UP}] PT_{ir}[\text{menit/UP}] + Y_{irt}^s[\] ST_{ir}[\text{menit}]$ $\leq C_i[\text{menit/UK}] QC_{it}^s[\text{UK}]$ $[\text{menit}] + [\text{menit}] \leq [\text{menit}]$ $[\text{menit}] \leq [\text{menit}] \quad \therefore \text{dimensi konsisten}$
(4.99)	$\frac{L_{rt}^s[\text{unit}]}{D_{rt}[\text{unit}]} \leq LL[\]$ $[\] \leq [\] \quad \therefore \text{dimensi konsisten}$
(4.100)	$Y_{irt}^s[\](AT_i[\text{menit}] + ST_{ir}[\text{menit}] + g_i[\text{km}] TT_i[\text{menit/km}])$ $+ X_{irt}^s[\text{UP}] PT_{ir}[\text{menit/UP}] \leq LT_r[\text{menit}]$ $[\]([\text{menit}] + [\text{menit}] + [\text{menit}]) + [\text{menit}] \leq [\text{menit}]$ $[\text{menit}] \leq [\text{menit}] \quad \therefore \text{dimensi konsisten}$

B. *Error Check dan Exception Handling*

Pemeriksaan kesalahan pemrograman atau *error checking* dan penanganan pengecualian atau *exception handling* dilakukan untuk memastikan bahwa tidak terdapat kesalahan sintaks dalam proses interpretasi model ke dalam bentuk kode pemrograman. Pemeriksaan kesalahan dan pengecualian dapat dilakukan menggunakan kode pemrograman berikut.

```
try:
    tr = gp.read("TRF.lp")
    tr.optimize()
except gp.GurobiError as e:
    print('Error code ' + str(e.errno) + ": " + str(e))
except AttributeError:
    print('Encountered an attribute error')
```

Dari hasil pemeriksaan kesalahan pemrograman, tidak ditemukan adanya pesan *gurobi error* maupun *attribute error*.

C. *Result Crosscheck*

Pemeriksaan silang hasil solusi atau *result crosscheck* dilakukan untuk mengetahui apakah solusi yang dihasilkan dari algoritma penyelesaian sudah sesuai dengan batasan model serta untuk menghindari pelanggaran batasan yang signifikan dari solusi yang dihasilkan. *Result crosscheck* dilakukan dengan menghitung secara manual kesesuaian solusi yang didapatkan dengan batasan yang ada. Tahap ini dilakukan dengan bantuan perangkat lunak Excel.

Berikut adalah ringkasan dari hasil pemeriksaan kesesuaian solusi terhadap batasan secara manual.

Tabel 4.5 Pemeriksaan Kesesuaian Solusi Terhadap Batasan

Batasan	<i>Expanded Constraint</i>	<i>Hasil Crosscheck</i>
(4.55)	6	Sesuai
(4.72)	15	Sesuai
(4.73)	15	Sesuai
(4.74)	15	Sesuai
(4.76)	15	Sesuai
(4.77)-(4.78)	250	Sesuai
(4.79)	275	Sesuai
(4.80)-(4.81)	1	Sesuai
(4.82)	250	Sesuai
(4.87)	6	Sesuai
(4.88)	300	Sesuai
(4.90)-(4.91)	300	Sesuai
(4.92)	5	Sesuai
(4.93)	300	Sesuai
(4.99)	250	Sesuai
(4.100)	1500	Sesuai
(4.101)	1500	Sesuai
(4.102)	1500	Sesuai

Dari hasil *crosscheck*, diketahui bahwa tidak terdapat batasan yang terlanggar oleh variabel keputusan yang dihasilkan oleh algoritma penyelesaian.

4.3.4 Perbaikan Penyelesaian dan *Final Testing*

Pengujian akhir atau *final testing* dilakukan untuk memastikan bahwa kode pemrograman yang sudah diperbaiki dapat berjalan dengan benar. Selain itu, untuk mengurangi waktu penyelesaian pemrograman model, dapat dilakukan penyesuaian terhadap parameter penyelesaian program (*running parameter*). Penyesuaian parameter penyelesaian program ini dilakukan bersamaan dengan *final testing* melalui *tuning*.

Parameter tuning dapat dilakukan secara otomatis dengan menggunakan *automatic tuning* maupun secara manual. Berikut adalah kode pemrograman untuk melakukan *automatic tuning*.

```

rt = gp.read("TRF.lp ")
rt.Params.TuneTimeLimit = 1800
rt.Params.TuneTrials = 3
rt.Params.TuneResults = -1
rt.Params.TuneCriterion = -1
rt.Params.TuneOutput = 3
rt.tune()
if rt.tuneResultCount > 0:
    rt.getTuneResult(0)
    rt.write('tuneTRF.prm')

```

Dilakukan *automatic tuning* selama 60 menit untuk setiap fungsi tujuan dalam model MO-MILP. Dari hasil eksekusi kode, didapatkan hasil *automatic tuning* seperti berikut.

Tabel 4.6 Hasil *Automatic Tuning*

Fungsi Tujuan	Parameter Set	MIP Gap	Mean Runtime	Parameter Penyelesaian
JTC	<i>Baseline</i>	0.07%	-	-
	<i>Improved set 1</i>	0.06%	-	<i>Presolve: 2</i>
TVP	<i>Baseline</i>	-	137.01 s	-
	<i>Improved set 1</i>	-	37.35 s	<i>GomoryPasses: 0</i>
GS	<i>Baseline</i>	-	0.93 s	-
	<i>Improved set 1</i>	-	0.73 s	<i>Method: 2</i>
	<i>Improved set 2</i>	-	0.44 s	<i>NormAdjust: 3</i> <i>PreDual: 1</i>
	<i>Improved set 3</i>	-	0.31 s	<i>Heuristics: 0.001</i> <i>PreDual: 1</i> <i>PrePasses: 1</i>
	<i>Improved set 4</i>	-	0.29 s	<i>Heuristics: 0.001</i> <i>Aggregate: 0</i> <i>PreDual: 1</i> <i>PrePasses: 1</i>

Karena *automatic tuning* tidak dapat meningkatkan efisiensi waktu penyelesaian pada fungsi transformasi, maka dilakukan *tuning* manual untuk fungsi tujuan transformasi. Berikut adalah hasil penentuan parameter penyelesaian pada fungsi tujuan transformasi.

Tabel 4.7 Hasil *Manual Tuning*

Fungsi Tujuan	Parameter Penyelesaian	Nilai	Deskripsi
Transformasi	Presolve	2	<i>Aggressive presolve</i>
	ScaleFlag	2	<i>Intermediate scaling</i>
	NodeMethod	2	<i>Barrier</i>

Sehingga, kode pemrograman untuk eksekusi penyelesaian model MO-MILP menjadi seperti berikut.

```
pr1 = gp.read("JTCmin.lp")
pr1.Params.Presolve = 2
pr1.optimize()
pr2 = gp.read("TVPmin.lp")
pr2.Params.GomoryPasses = 0
pr2.optimize()
pr3 = gp.read("TVPmax.lp")
pr3.Params.GomoryPasses = 0
pr3.optimize()
pr4 = gp.read("GSmin.lp")
pr4.Params.Heuristics = 0.001
pr4.Params.Aggregate = 0
pr4.Params.PreDual = 1
pr4.Params.PrePasses = 1
pr4.optimize()
pr5 = gp.read("GSmax.lp")
pr5.Params.Heuristics = 0.001
pr5.Params.Aggregate = 0
pr5.Params.PreDual = 1
pr5.Params.PrePasses = 1
pr5.optimize()
tr = gp.read("TRF.lp")
tr.Params.Presolve = 2
tr.Params.ScaleFlag = 2
tr.Params.NodeMethod = 2
tr.optimize()
```

4.4 CONTOH NUMERIK

Bagian ini menjelaskan mengenai implementasi kerangka kerja model melalui contoh numerik. Contoh numerik yang dilakukan dimulai dari pengolahan rating pemasok hingga pada pengolahan model optimasi pemilihan pemasok dan alokasi pesanan.

4.4.1 Pengolahan *Rating* Pemasok

Dalam pengolahan *rating* pemasok, data hasil penilaian hubungan pengaruh langsung (*direct-influence*) antar setiap kriteria dan data penilaian kinerja pemasok didasarkan pada penelitian Goren (2018).

Berikut adalah hasil matriks pengaruh langsung (*direct-influential matrix*) dalam variabel linguistik berdasarkan pada penelitian Goren (2018).

Tabel 4.8 Matriks Pengaruh Langsung Linguistik

$D_u \backslash D_v$		D_1								D_2		D_3		
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
D_1	C_1	N	N	VL	VL	H	H	VL	H	N	L	VL	L	VL
	C_2	H	N	N	H	H	H	N	H	N	VL	N	VL	H
	C_3	N	N	N	L	VH	N	N	VH	N	N	N	N	H
	C_4	L	L	VL	N	H	N	N	H	VL	L	VL	VL	L
	C_5	N	N	N	H	N	VL	N	VL	N	N	VL	L	H
	C_6	H	N	N	H	VL	N	N	N	N	L	L	VL	VL
	C_7	H	H	VH	H	L	VH	N	L	L	L	L	VL	VH
	C_8	N	N	N	VH	N	N	N	N	N	N	N	N	N
D_2	C_9	VL	VL	N	L	N	VL	N	N	N	VL	VL	VL	L
	C_{10}	VL	VL	N	L	N	VL	N	N	VL	N	VL	VL	VL
D_3	C_{11}	VL	N	N	H	N	L	N	N	H	L	N	L	H
	C_{12}	H	VL	N	H	L	VL	N	N	L	VL	VL	N	H
	C_{13}	H	H	L	L	VL	VL	N	N	VL	N	H	VL	N

Selanjutnya, matriks pengaruh langsung (*direct-influential matrix*) dalam skala linguistik diinterpretasikan dalam bentuk bilangan *fuzzy*. Berikut adalah hasil matriks pengaruh langsung (*direct-influential matrix*) dalam bilangan *fuzzy* berdasarkan pada konversi skala linguistik Li (1999).

Tabel 4.9 Matriks Pengaruh Langsung *Fuzzy*

$D_u \backslash D_v$		D_1						
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5	C_6	C_7
D_1	C_1	(0,0,0.25)	(0,0,0.25)	(0,0.25,0.5)	(0,0.25,0.5)	(0.5,0.75,1)	(0.5,0.75,1)	(0,0.25,0.5)
	C_2	(0.5,0.75,1)	(0,0,0.25)	(0,0,0.25)	(0.5,0.75,1)	(0.5,0.75,1)	(0.5,0.75,1)	(0,0,0.25)
	C_3	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	(0.25,0.5,0.75)	(0.75,1,1)	(0,0,0.25)	(0,0,0.25)
	C_4	(0.25,0.5,0.75)	(0.25,0.5,0.75)	(0,0.25,0.5)	(0,0,0.25)	(0.5,0.75,1)	(0,0,0.25)	(0,0,0.25)
	C_5	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	(0.5,0.75,1)	(0,0,0.25)	(0,0.25,0.5)	(0,0,0.25)
	C_6	(0.5,0.75,1)	(0,0,0.25)	(0,0,0.25)	(0.5,0.75,1)	(0,0.25,0.5)	(0,0,0.25)	(0,0,0.25)
	C_7	(0.5,0.75,1)	(0.5,0.75,1)	(0.75,1,1)	(0.5,0.75,1)	(0.25,0.5,0.75)	(0.75,1,1)	(0,0,0.25)
	C_8	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	(0.75,1,1)	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)
D_2	C_9	(0,0.25,0.5)	(0,0.25,0.5)	(0,0,0.25)	(0.25,0.5,0.75)	(0,0,0.25)	(0,0.25,0.5)	(0,0,0.25)
	C_{10}	(0,0.25,0.5)	(0,0.25,0.5)	(0,0,0.25)	(0.25,0.5,0.75)	(0,0,0.25)	(0,0.25,0.5)	(0,0,0.25)
D_3	C_{11}	(0,0.25,0.5)	(0,0,0.25)	(0,0,0.25)	(0.5,0.75,1)	(0,0,0.25)	(0.25,0.5,0.75)	(0,0,0.25)
	C_{12}	(0.5,0.75,1)	(0,0.25,0.5)	(0,0,0.25)	(0.5,0.75,1)	(0.25,0.5,0.75)	(0,0.25,0.5)	(0,0,0.25)
	C_{13}	(0.5,0.75,1)	(0.5,0.75,1)	(0.25,0.5,0.75)	(0.25,0.5,0.75)	(0,0.25,0.5)	(0,0.25,0.5)	(0,0,0.25)

Tabel 4.9 Matriks Pengaruh Langsung Fuzzy (lanjutan)

$D_u \backslash D_v$		D_1	D_2		D_3			$\Sigma z_{(r)ij}$
	$C_i \backslash C_j$	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	
D_1	C_1	(0.5,0.75,1)	(0,0,0.25)	(0.25,0.5,0.75)	(0,0.25,0.5)	(0.25,0.5,0.75)	(0,0.25,0.5)	7.75
	C_2	(0.5,0.75,1)	(0,0,0.25)	(0,0.25,0.5)	(0,0,0.25)	(0,0.25,0.5)	(0.5,0.75,1)	8.25
	C_3	(0.75,1,1)	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	(0.5,0.75,1)	6
	C_4	(0.5,0.75,1)	(0,0.25,0.5)	(0.25,0.5,0.75)	(0,0.25,0.5)	(0,0.25,0.5)	(0.25,0.5,0.75)	7.75
	C_5	(0,0.25,0.5)	(0,0,0.25)	(0,0,0.25)	(0,0.25,0.5)	(0.25,0.5,0.75)	(0.5,0.75,1)	6
	C_6	(0,0,0.25)	(0,0,0.25)	(0.25,0.5,0.75)	(0.25,0.5,0.75)	(0,0.25,0.5)	(0,0.25,0.5)	6.5
	C_7	(0.25,0.5,0.75)	(0.25,0.5,0.75)	(0.25,0.5,0.75)	(0.25,0.5,0.75)	(0,0.25,0.5)	(0.75,1,1)	10.5
	C_8	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	(0,0,0.25)	4
D_2	C_9	(0,0,0.25)	(0,0,0.25)	(0,0.25,0.5)	(0,0.25,0.5)	(0,0.25,0.5)	(0.25,0.5,0.75)	5.75
	C_{10}	(0,0,0.25)	(0,0.25,0.5)	(0,0,0.25)	(0,0.25,0.5)	(0,0.25,0.5)	(0,0.25,0.5)	5.5
D_3	C_{11}	(0,0,0.25)	(0.5,0.75,1)	(0.25,0.5,0.75)	(0,0,0.25)	(0.25,0.5,0.75)	(0.5,0.75,1)	7.25
	C_{12}	(0,0,0.25)	(0.25,0.5,0.75)	(0,0.25,0.5)	(0,0.25,0.5)	(0,0,0.25)	(0.5,0.75,1)	7.5
	C_{13}	(0,0,0.25)	(0,0.25,0.5)	(0,0,0.25)	(0.5,0.75,1)	(0,0.25,0.5)	(0,0,0.25)	7.5
$\max(\Sigma z_{(r)ij})$								10.5
λ								0.09524

Kemudian, dilakukan proses normalisasi terhadap matriks pengaruh langsung (*direct-influential matrix*) fuzzy. Berikut adalah hasil perhitungan matriks pengaruh langsung ternormalisasi (*normalized direct-influential matrix*).

Tabel 4.10 Matriks Pengaruh Langsung Fuzzy Ternormalisasi

$D_u \backslash D_v$		D_1				
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5
D_1	C_1	(0,0,0.024)	(0,0,0.024)	(0,0.024,0.048)	(0,0.024,0.048)	(0.048,0.071,0.095)
	C_2	(0.048,0.071,0.095)	(0,0,0.024)	(0,0,0.024)	(0.048,0.071,0.095)	(0.048,0.071,0.095)
	C_3	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	(0.024,0.048,0.071)	(0.071,0.095,0.095)
	C_4	(0.024,0.048,0.071)	(0.024,0.048,0.071)	(0,0.024,0.048)	(0,0,0.024)	(0.048,0.071,0.095)
	C_5	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	(0.048,0.071,0.095)	(0,0,0.024)
	C_6	(0.048,0.071,0.095)	(0,0,0.024)	(0,0,0.024)	(0.048,0.071,0.095)	(0,0.024,0.048)
	C_7	(0.048,0.071,0.095)	(0.048,0.071,0.095)	(0.071,0.095,0.095)	(0.048,0.071,0.095)	(0.024,0.048,0.071)
	C_8	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	(0.071,0.095,0.095)	(0,0,0.024)
D_2	C_9	(0,0.024,0.048)	(0,0.024,0.048)	(0,0,0.024)	(0.024,0.048,0.071)	(0,0,0.024)
	C_{10}	(0,0.024,0.048)	(0,0.024,0.048)	(0,0,0.024)	(0.024,0.048,0.071)	(0,0,0.024)
D_3	C_{11}	(0,0.024,0.048)	(0,0,0.024)	(0,0,0.024)	(0.048,0.071,0.095)	(0,0,0.024)
	C_{12}	(0.048,0.071,0.095)	(0,0.024,0.048)	(0,0,0.024)	(0.048,0.071,0.095)	(0.024,0.048,0.071)
	C_{13}	(0.048,0.071,0.095)	(0.048,0.071,0.095)	(0.024,0.048,0.071)	(0.024,0.048,0.071)	(0,0.024,0.048)

Tabel 4.10 Matriks Pengaruh Langsung *Fuzzy* Ternormalisasi (lanjutan)

$D_u \backslash D_v$		D_1			D_2	
	$C_i \backslash C_j$	C_6	C_7	C_8	C_9	C_{10}
D_1	C_1	(0.048,0.071,0.095)	(0,0.024,0.048)	(0.048,0.071,0.095)	(0,0,0.024)	(0.024,0.048,0.071)
	C_2	(0.048,0.071,0.095)	(0,0,0.024)	(0.048,0.071,0.095)	(0,0,0.024)	(0,0.024,0.048)
	C_3	(0,0,0.024)	(0,0,0.024)	(0.071,0.095,0.095)	(0,0,0.024)	(0,0,0.024)
	C_4	(0,0,0.024)	(0,0,0.024)	(0.048,0.071,0.095)	(0,0.024,0.048)	(0.024,0.048,0.071)
	C_5	(0,0.024,0.048)	(0,0,0.024)	(0,0.024,0.048)	(0,0,0.024)	(0,0,0.024)
	C_6	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	(0.024,0.048,0.071)
	C_7	(0.071,0.095,0.095)	(0,0,0.024)	(0.024,0.048,0.071)	(0.024,0.048,0.071)	(0.024,0.048,0.071)
	C_8	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)
D_2	C_9	(0,0.024,0.048)	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	(0,0.024,0.048)
	C_{10}	(0,0.024,0.048)	(0,0,0.024)	(0,0,0.024)	(0,0.024,0.048)	(0,0,0.024)
D_3	C_{11}	(0.024,0.048,0.071)	(0,0,0.024)	(0,0,0.024)	(0.048,0.071,0.095)	(0.024,0.048,0.071)
	C_{12}	(0,0.024,0.048)	(0,0,0.024)	(0,0,0.024)	(0.024,0.048,0.071)	(0,0.024,0.048)
	C_{13}	(0,0.024,0.048)	(0,0,0.024)	(0,0,0.024)	(0,0.024,0.048)	(0,0,0.024)

Tabel 4.10 Matriks Pengaruh Langsung *Fuzzy* Ternormalisasi (lanjutan)

$D_u \backslash D_v$		D_3			$\Sigma x_{(r)ij}$
	$C_i \backslash C_j$	C_{11}	C_{12}	C_{13}	
D_1	C_1	(0,0.024,0.048)	(0.024,0.048,0.071)	(0,0.024,0.048)	0.7381
	C_2	(0,0,0.024)	(0,0.024,0.048)	(0.048,0.071,0.095)	0.7857
	C_3	(0,0,0.024)	(0,0,0.024)	(0.048,0.071,0.095)	0.5714
	C_4	(0,0.024,0.048)	(0,0.024,0.048)	(0.024,0.048,0.071)	0.7381
	C_5	(0,0.024,0.048)	(0.024,0.048,0.071)	(0.048,0.071,0.095)	0.5714
	C_6	(0.024,0.048,0.071)	(0,0.024,0.048)	(0,0.024,0.048)	0.6190
	C_7	(0.024,0.048,0.071)	(0,0.024,0.048)	(0.071,0.095,0.095)	1.0000
	C_8	(0,0,0.024)	(0,0,0.024)	(0,0,0.024)	0.3810
D_2	C_9	(0,0.024,0.048)	(0,0.024,0.048)	(0.024,0.048,0.071)	0.5476
	C_{10}	(0,0.024,0.048)	(0,0.024,0.048)	(0,0.024,0.048)	0.5238
D_3	C_{11}	(0,0,0.024)	(0.024,0.048,0.071)	(0.048,0.071,0.095)	0.6905
	C_{12}	(0,0.024,0.048)	(0,0,0.024)	(0.048,0.071,0.095)	0.7143
	C_{13}	(0.048,0.071,0.095)	(0,0.024,0.048)	(0,0,0.024)	0.7143
$\max(\Sigma x_{(r)ij})$					1.000

Lalu, dilakukan perhitungan matriks pengaruh total (*total-influential matrix*) fuzzy dari hasil matriks pengaruh langsung ternormalisasi. Berikut adalah hasil perhitungan matriks pengaruh total (*total-influential matrix*).

Tabel 4.11 Matriks Pengaruh Total Fuzzy

$D_u \backslash D_v$		D_1				
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5
D_1	C_1	(0.0039,0.0222,0.1490)	(0.0004,0.0115,0.1149)	(0.0001,0.0305,0.1221)	(0.0100,0.0618,0.2106)	(0.0489,0.0885,0.2098)
	C_2	(0.0542,0.0955,0.2288)	(0.0039,0.0143,0.1231)	(0.0012,0.0096,0.1064)	(0.0582,0.1083,0.2647)	(0.0533,0.0944,0.2224)
	C_3	(0.0036,0.0122,0.1190)	(0.0033,0.0102,0.0974)	(0.0012,0.0062,0.0839)	(0.0342,0.0732,0.1945)	(0.0735,0.1054,0.1842)
	C_4	(0.0268,0.0659,0.1927)	(0.0253,0.0578,0.1612)	(0.0007,0.0299,0.1219)	(0.0088,0.0366,0.1852)	(0.0506,0.0898,0.2111)
	C_5	(0.0050,0.0192,0.1294)	(0.0036,0.0124,0.1024)	(0.0012,0.0068,0.0868)	(0.0507,0.0915,0.2217)	(0.0035,0.0150,0.1209)
	C_6	(0.0492,0.0868,0.2027)	(0.0013,0.0103,0.1049)	(0.0001,0.0066,0.0907)	(0.0504,0.0927,0.2301)	(0.0049,0.0413,0.1528)
	C_7	(0.0595,0.1123,0.2671)	(0.0531,0.0926,0.2201)	(0.0734,0.1081,0.1983)	(0.0649,0.1319,0.3134)	(0.0376,0.0909,0.2375)
	C_8	(0.0019,0.0063,0.0920)	(0.0018,0.0055,0.0753)	(0.0000,0.0028,0.0655)	(0.0721,0.0987,0.1777)	(0.0036,0.0086,0.0904)
D_2	C_9	(0.0019,0.0405,0.1483)	(0.0018,0.0333,0.1212)	(0.0006,0.0056,0.0838)	(0.0247,0.0657,0.1936)	(0.0014,0.0145,0.1181)
	C_{10}	(0.0006,0.0384,0.1432)	(0.0006,0.0314,0.1169)	(0.0000,0.0044,0.0804)	(0.0240,0.0638,0.1882)	(0.0012,0.0133,0.1141)
D_3	C_{11}	(0.0062,0.0490,0.1727)	(0.0037,0.0163,0.1172)	(0.0012,0.0080,0.0978)	(0.0537,0.0982,0.2450)	(0.0037,0.0191,0.1380)
	C_{12}	(0.0517,0.0936,0.2209)	(0.0037,0.0384,0.1422)	(0.0012,0.0094,0.1017)	(0.0518,0.1003,0.2511)	(0.0290,0.0694,0.1917)
	C_{13}	(0.0514,0.0928,0.2205)	(0.0487,0.0799,0.1842)	(0.0240,0.0535,0.1462)	(0.0306,0.0803,0.2315)	(0.0080,0.0509,0.1726)

Tabel 4.11 Matriks Pengaruh Total *Fuzzy* (lanjutan)

$D_u \backslash D_v$		D_1			D_2
	$C_i \backslash C_j$	C_6	C_7	C_8	C_9
D_1	C_1	(0.0479,0.0845,0.1984)	(0.0000,0.0243,0.1009)	(0.0483,0.0844,0.1974)	(0.0007,0.0111,0.1107)
	C_2	(0.0505,0.0872,0.2073)	(0.0000,0.0023,0.0827)	(0.0532,0.0903,0.2096)	(0.0002,0.0094,0.1141)
	C_3	(0.0004,0.0072,0.1033)	(0.0000,0.0003,0.0651)	(0.0735,0.1052,0.1765)	(0.0002,0.0053,0.0893)
	C_4	(0.0025,0.0177,0.1321)	(0.0000,0.0016,0.0781)	(0.0506,0.0879,0.2009)	(0.0001,0.0324,0.1320)
	C_5	(0.0005,0.0321,0.1315)	(0.0000,0.0005,0.0670)	(0.0029,0.0336,0.1323)	(0.0007,0.0098,0.0964)
	C_6	(0.0030,0.0146,0.1191)	(0.0000,0.0021,0.0720)	(0.0048,0.0153,0.1191)	(0.0012,0.0109,0.1019)
	C_7	(0.0775,0.1238,0.2420)	(0.0000,0.0027,0.0987)	(0.0375,0.0843,0.2217)	(0.0253,0.0640,0.1879)
	C_8	(0.0002,0.0017,0.0800)	(0.0000,0.0001,0.0528)	(0.0036,0.0084,0.0846)	(0.0000,0.0031,0.0709)
D_2	C_9	(0.0002,0.0345,0.1316)	(0.0000,0.0010,0.0660)	(0.0014,0.0109,0.1075)	(0.0001,0.0080,0.0926)
	C_{10}	(0.0001,0.0334,0.1279)	(0.0000,0.0009,0.0641)	(0.0012,0.0103,0.1043)	(0.0000,0.0304,0.1127)
D_3	C_{11}	(0.0244,0.0608,0.1723)	(0.0000,0.0012,0.0758)	(0.0031,0.0130,0.1238)	(0.0483,0.0818,0.1789)
	C_{12}	(0.0027,0.0421,0.1568)	(0.0000,0.0022,0.0790)	(0.0052,0.0192,0.1332)	(0.0240,0.0572,0.1571)
	C_{13}	(0.0059,0.0445,0.1597)	(0.0000,0.0022,0.0793)	(0.0079,0.0245,0.1378)	(0.0024,0.0350,0.1353)

Tabel 4.11 Matriks Pengaruh Total *Fuzzy* (lanjutan)

$D_u \backslash D_v$		D_2	D_3		
	$C_i \backslash C_j$	C_{10}	C_{11}	C_{12}	C_{13}
D_1	C_1	(0.0253,0.0607,0.1692)	(0.0013,0.0399,0.1493)	(0.0251,0.0620,0.1714)	(0.0039,0.0502,0.1870)
	C_2	(0.0040,0.0403,0.1542)	(0.0037,0.0204,0.1354)	(0.0026,0.0423,0.1573)	(0.0521,0.0955,0.2399)
	C_3	(0.0010,0.0056,0.0970)	(0.0025,0.0116,0.1033)	(0.0019,0.0106,0.1018)	(0.0524,0.0861,0.2002)
	C_4	(0.0247,0.0583,0.1667)	(0.0014,0.0376,0.1470)	(0.0019,0.0396,0.1489)	(0.0278,0.0726,0.2100)
	C_5	(0.0014,0.0104,0.1047)	(0.0024,0.0364,0.1315)	(0.0241,0.0568,0.1518)	(0.0505,0.0869,0.2063)
	C_6	(0.0268,0.0612,0.1603)	(0.0240,0.0594,0.1595)	(0.0019,0.0384,0.1383)	(0.0028,0.0440,0.1696)
	C_7	(0.0293,0.0739,0.2080)	(0.0296,0.0763,0.2101)	(0.0030,0.0521,0.1873)	(0.0829,0.1422,0.2870)
	C_8	(0.0018,0.0056,0.0788)	(0.0001,0.0036,0.0780)	(0.0001,0.0038,0.0781)	(0.0020,0.0069,0.0998)
D_2	C_9	(0.0007,0.0340,0.1261)	(0.0012,0.0346,0.1276)	(0.0001,0.0338,0.1267)	(0.0246,0.0625,0.1793)
	C_{10}	(0.0006,0.0103,0.0999)	(0.0000,0.0327,0.1231)	(0.0000,0.0328,0.1232)	(0.0007,0.0385,0.1521)
D_3	C_{11}	(0.0259,0.0623,0.1680)	(0.0030,0.0185,0.1252)	(0.0241,0.0616,0.1682)	(0.0518,0.0930,0.2271)
	C_{12}	(0.0026,0.0399,0.1486)	(0.0025,0.0418,0.1516)	(0.0020,0.0187,0.1284)	(0.0513,0.0960,0.2339)
	C_{13}	(0.0032,0.0182,0.1274)	(0.0481,0.0835,0.1937)	(0.0026,0.0416,0.1517)	(0.0070,0.0315,0.1698)

Kemudian, dilakukan proses defuzzifikasi (*defuzzification*) pada matriks pengaruh total (*total-influential matrix*) fuzzy untuk mengubah bilangan fuzzy menjadi bilangan *crisp*. Proses defuzzifikasi ini dilakukan menggunakan metode CFCS (*Converting Fuzzy Data into Crisp Scores*). Berikut adalah hasil defuzzifikasi matriks pengaruh total kriteria.

Tabel 4.12 Matriks Pengaruh Total Kriteria

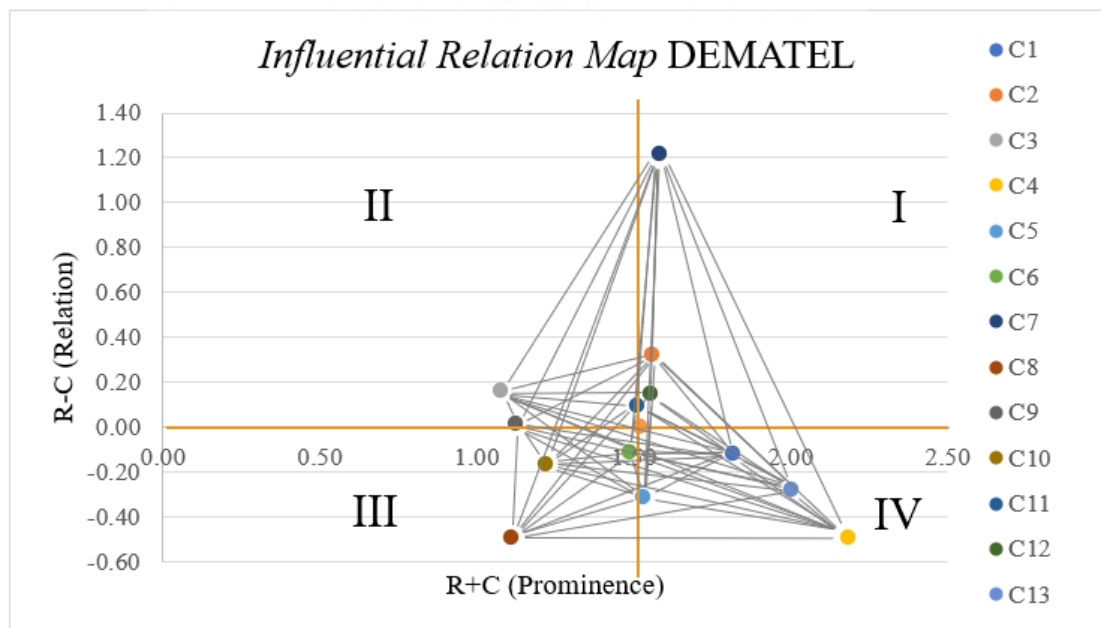
$D_u \backslash D_v$		D_1								D_2		D_3		
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
D_1	C_1	0.0443	0.0293	0.0451	0.0816	0.1035	0.0993	0.0365	0.0992	0.0282	0.0766	0.0568	0.0778	0.0701
	C_2	0.1143	0.0327	0.0253	0.1279	0.1124	0.1050	0.0142	0.1078	0.0267	0.0586	0.0401	0.0605	0.1157
	C_3	0.0309	0.0251	0.0189	0.0892	0.1135	0.0237	0.0099	0.1122	0.0192	0.0211	0.0274	0.0263	0.1007
	C_4	0.0839	0.0734	0.0447	0.0612	0.1048	0.0377	0.0135	0.1022	0.0484	0.0746	0.0548	0.0566	0.0907
	C_5	0.0384	0.0276	0.0197	0.1079	0.0334	0.0481	0.0101	0.0496	0.0242	0.0264	0.0513	0.0711	0.1030
	C_6	0.1026	0.0262	0.0201	0.1103	0.0589	0.0326	0.0124	0.0332	0.0261	0.0761	0.0744	0.0539	0.0635
	C_7	0.1334	0.1122	0.1209	0.1529	0.1119	0.1392	0.0169	0.1049	0.0837	0.0946	0.0966	0.0732	0.1582
	C_8	0.0210	0.0170	0.0127	0.1071	0.0227	0.0146	0.0077	0.0214	0.0140	0.0178	0.0159	0.0160	0.0230
D_2	C_9	0.0568	0.0470	0.0185	0.0826	0.0325	0.0496	0.0108	0.0276	0.0223	0.0483	0.0491	0.0482	0.0786
	C_{10}	0.0542	0.0448	0.0168	0.0803	0.0308	0.0481	0.0105	0.0266	0.0432	0.0257	0.0468	0.0468	0.0558
D_3	C_{11}	0.0681	0.0334	0.0226	0.1172	0.0397	0.0779	0.0121	0.0320	0.0958	0.0783	0.0367	0.0776	0.1112
	C_{12}	0.1109	0.0549	0.0245	0.1193	0.0876	0.0602	0.0136	0.0389	0.0728	0.0571	0.0590	0.0374	0.1142
	C_{13}	0.1093	0.0954	0.0680	0.1000	0.0692	0.0625	0.0140	0.0443	0.0510	0.0371	0.0990	0.0587	0.0552

Selanjutnya, dapat dilakukan pembentukan peta hubungan pengaruh atau *Influential Relation Map* (IRM) antar kriteria berdasarkan metode DEMATEL. Berikut adalah hasil perhitungan *Influential Relation Map* (IRM) berdasarkan metode DEMATEL.

Tabel 4.13 Perhitungan *Influential Relation Map* (IRM) Kriteria Berdasarkan Metode DEMATEL

D_u	C_i	\vec{C}	\vec{R}	$\vec{R} + \vec{C}$	$\vec{R} - \vec{C}$	$L/R\ Side$	$U/D\ Side$	Kuadran
D_1	C_1	0.96794	0.84840	1.82	-0.12	Right	Down	IV
	C_2	0.61900	0.94119	1.56	0.32	Right	Up	I
	C_3	0.45801	0.61796	1.08	0.16	Left	Up	II
	C_4	1.33765	0.84652	2.18	-0.49	Right	Down	IV
	C_5	0.92103	0.61076	1.53	-0.31	Right	Down	IV
	C_6	0.79853	0.69027	1.49	-0.11	Left	Down	III
	C_7	0.18237	1.39860	1.58	1.22	Right	Up	I
	C_8	0.79979	0.31084	1.11	-0.49	Left	Down	III
D_2	C_9	0.55553	0.57189	1.13	0.02	Left	Up	II
	C_{10}	0.69208	0.53053	1.22	-0.16	Left	Down	III
D_3	C_{11}	0.70759	0.80264	1.51	0.10	Left	Up	II
	C_{12}	0.70418	0.85031	1.55	0.15	Right	Up	I
	C_{13}	1.13988	0.86368	2.00	-0.28	Right	Down	IV
Rata-rata				1.52	0.00			

Berikut adalah peta hubungan pengaruh atau *Influential Relation Map* (IRM) berdasarkan metode DEMATEL.



Gambar 4.6 *Influential Relation Map* (IRM) Kriteria Berdasarkan Metode DEMATEL

Lalu, dapat dilakukan perhitungan matriks pengaruh bersih (*net influence*) dari matriks pengaruh total. Berikut adalah hasil perhitungan matriks pengaruh bersih (*net influence matrix*).

Tabel 4.14 Matriks Pengaruh Bersih (*Net Influence*)

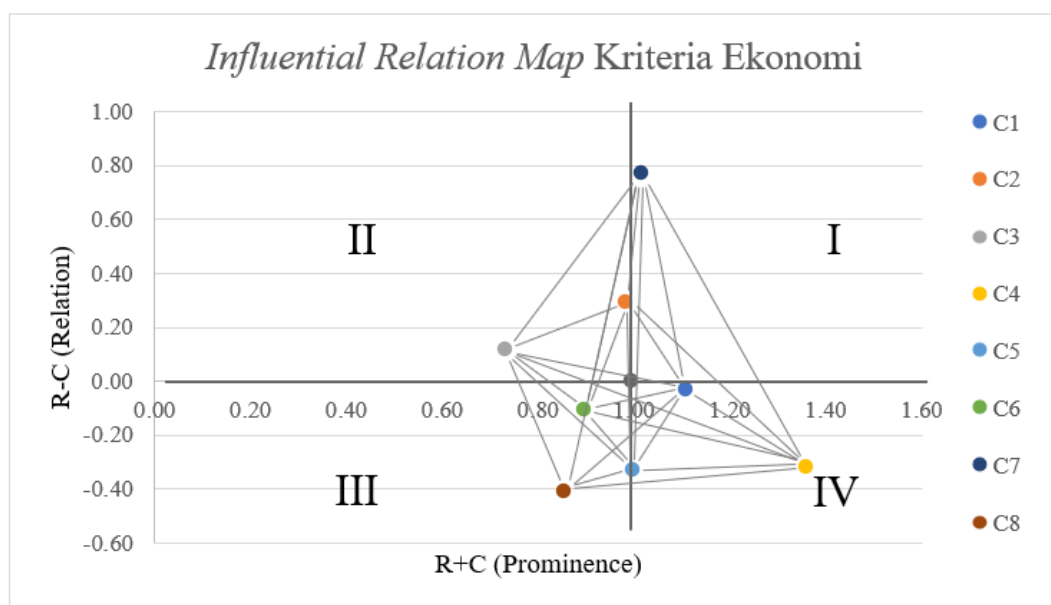
$D_u \backslash D_v$		D_1								D_2		D_3		
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
D_1	C_1	0	-0.085	0.014	-0.002	0.065	-0.003	-0.097	0.078	-0.029	0.022	-0.011	-0.033	-0.039
	C_2	0.085	0	0	0.055	0.085	0.079	-0.098	0.091	-0.020	0.014	0.007	0.006	0.02
	C_3	-0.014	0.000	0	0.044	0.094	0.004	-0.111	0.099	0.001	0.004	0.005	0.002	0.033
	C_4	0.002	-0.055	-0.044	0	-0.003	-0.073	-0.139	-0.005	-0.034	-0.006	-0.062	-0.063	-0.009
	C_5	-0.065	-0.085	-0.094	0.003	0	-0.011	-0.102	0.027	-0.008	-0.004	0.012	-0.016	0.034
	C_6	0.003	-0.079	-0.004	0.073	0.011	0	-0.127	0.019	-0.024	0.028	-0.004	-0.006	0.001
	C_7	0.097	0.098	0.111	0.139	0.102	0.127	0	0.097	0.073	0.084	0.084	0.060	0.144
	C_8	-0.078	-0.091	-0.099	0.005	-0.027	-0.019	-0.097	0	-0.014	-0.009	-0.016	-0.023	-0.021
D_2	C_9	0.029	0.020	-0.001	0.034	0.008	0.024	-0.073	0.014	0	0.005	-0.047	-0.025	0.028
	C_{10}	-0.022	-0.014	-0.004	0.006	0.004	-0.028	-0.084	0.009	-0.005	0	-0.032	-0.01	0.019
D_3	C_{11}	0.011	-0.007	-0.005	0.062	-0.012	0.004	-0.084	0.016	0.047	0.032	0	0.019	0.012
	C_{12}	0.033	-0.006	-0.002	0.063	0.016	0.006	-0.060	0.023	0.025	0.01	-0.019	0	0.056
	C_{13}	0.039	-0.020	-0.033	0.009	-0.034	-0.001	-0.144	0.021	-0.028	-0.019	-0.012	-0.056	0

Selanjutnya, dapat dilakukan pembentukan peta hubungan pengaruh atau *Influential Relation Map* (IRM) antar kriteria dalam dimensi berdasarkan metode DANP. Berikut adalah hasil perhitungan *Influential Relation Map* (IRM) kriteria dalam dimensi berdasarkan metode DANP.

Tabel 4.15 Perhitungan *Influential Relation Map* (IRM) Kriteria dalam Dimensi Berdasarkan Metode DANP

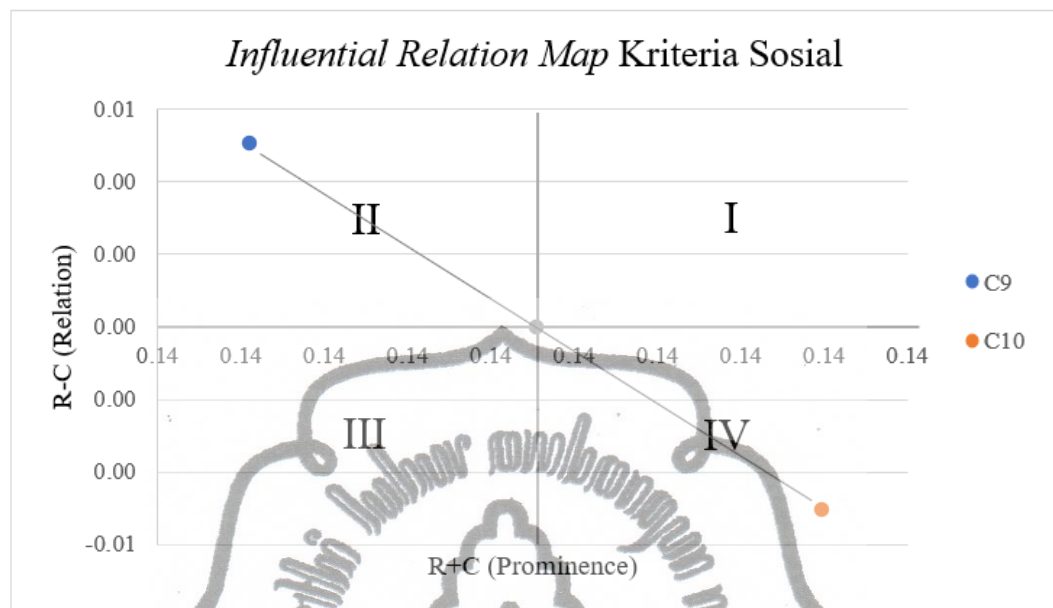
D_u	C_i	\vec{C}	\vec{R}	$\vec{R} + \vec{C}$	$\vec{R} - \vec{C}$	$L/R Side$	$U/D Side$	Kuadran
D_1	C_1	0.568771	0.53892	1.11	-0.03	Right	Down	IV
	C_2	0.343587	0.63971	0.98	0.30	Left	Up	II
	C_3	0.307553	0.42337	0.73	0.12	Left	Up	II
	C_4	0.838186	0.52144	1.36	-0.32	Right	Down	IV
	C_5	0.661101	0.33483	1.00	-0.33	Right	Down	IV
	C_6	0.500189	0.39634	0.90	-0.10	Left	Down	III
	C_7	0.121304	0.89234	1.01	0.77	Right	Up	I
	C_8	0.630416	0.22415	0.85	-0.41	Left	Down	III
D_2	C_9	0.065523	0.07058	0.14	0.01	Left	Up	II
	C_{10}	0.07402	0.06896	0.14	-0.01	Right	Down	IV
D_3	C_{11}	0.194683	0.22553	0.42	0.03	Left	Up	II
	C_{12}	0.173753	0.21061	0.38	0.04	Left	Up	II
	C_{13}	0.280642	0.21294	0.49	-0.07	Right	Down	IV
Rata-rata dalam D_1				0.99	0.00			
Rata-rata dalam D_2				0.14	0.00			
Rata-rata dalam D_3				0.43	0.00			

Berikut adalah peta hubungan pengaruh atau *Influential Relation Map* (IRM) antar kriteria dalam Dimensi Ekonomi berdasarkan metode DANP.



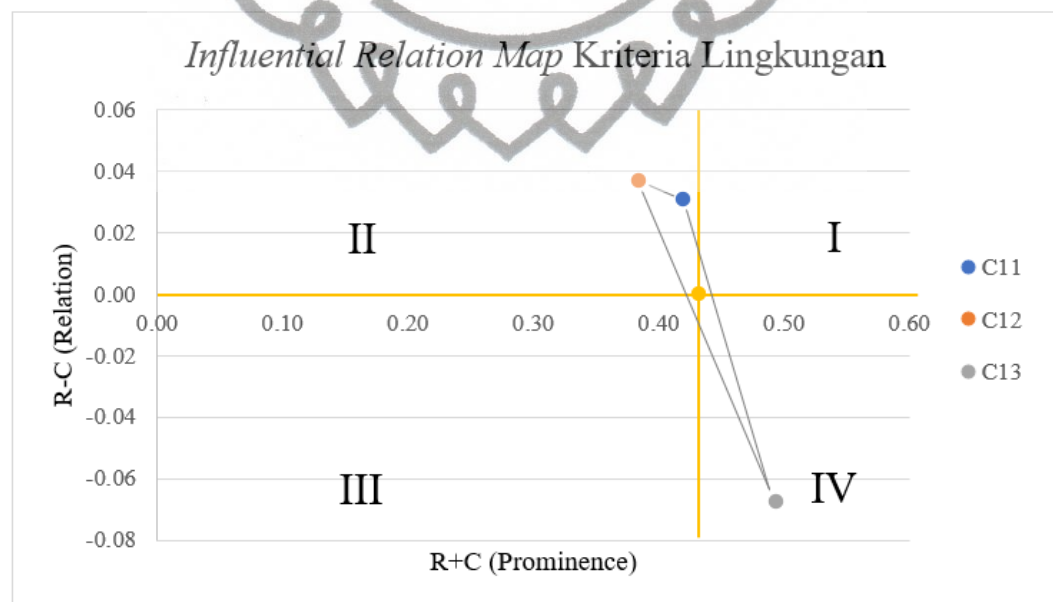
Gambar 4.7 *Influential Relation Map* (IRM) Kriteria dalam Dimensi Ekonomi

Berikut adalah peta hubungan pengaruh atau *Influential Relation Map* (IRM) antar kriteria dalam Dimensi Sosial berdasarkan metode DANP.



Gambar 4.8 *Influential Relation Map* (IRM) Kriteria dalam Dimensi Sosial

Berikut adalah peta hubungan pengaruh atau *Influential Relation Map* (IRM) antar kriteria dalam Dimensi Lingkungan berdasarkan metode DANP.



Gambar 4.9 *Influential Relation Map* (IRM) Kriteria dalam Dimensi Lingkungan

Kemudian, dilakukan perhitungan matriks pengaruh total dimensi berdasarkan matriks pengaruh total kriteria dengan nilai *crisp*. Berikut adalah hasil perhitungan dari matriks pengaruh total dimensi.

Tabel 4.16 Matriks Pengaruh Total Dimensi

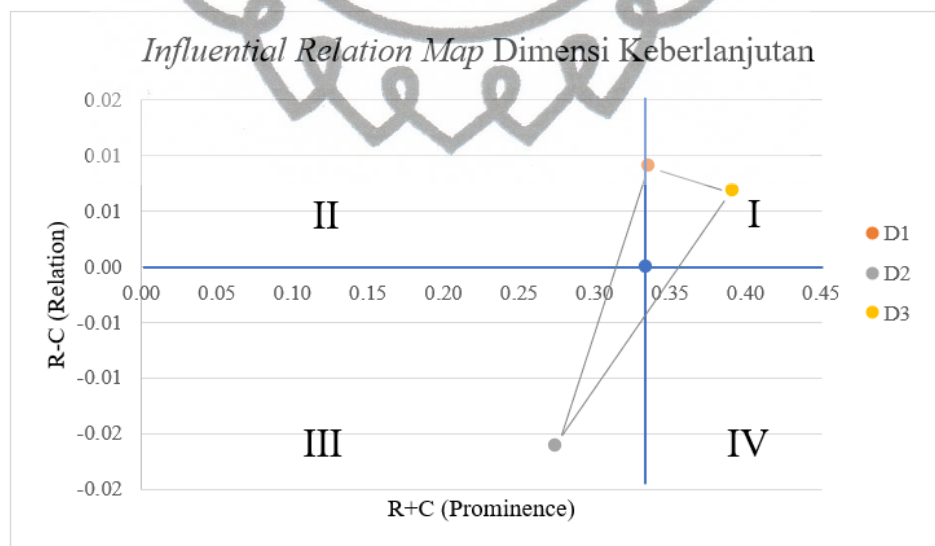
D_u	D_1	D_2	D_3
D_1	0.062049	0.044755	0.065723
D_2	0.039853	0.034886	0.054205
D_3	0.061482	0.065331	0.07212

Selanjutnya, dapat dilakukan pembentukan peta hubungan pengaruh atau *Influential Relation Map* (IRM) antar dimensi berdasarkan metode DANP. Berikut adalah hasil perhitungan *Influential Relation Map* (IRM) kriteria berdasarkan metode DANP.

Tabel 4.17 Perhitungan *Influential Relation Map* (IRM) Dimensi Berdasarkan Metode DANP

D_u	\bar{C}	\bar{R}	$\bar{R} + \bar{C}$	$\bar{R} - \bar{C}$	L/R Side	U/D Side	Kuadran
D_1	0.16338	0.17253	0.34	0.01	Right	Up	I
D_2	0.14497	0.12894	0.27	-0.02	Left	Down	III
D_3	0.19205	0.19893	0.39	0.01	Right	Up	I
Rata-rata			0.33	0.00			

Berikut adalah peta hubungan pengaruh atau *Influential Relation Map* (IRM) antar dimensi berdasarkan metode DANP.

**Gambar 4.10** *Influential Relation Map* (IRM) Dimensi Keberlanjutan

Kemudian, dapat dilakukan proses normalisasi pada matriks pengaruh total (*total-influential matrix*) kriteria berdasarkan dimensinya. Berikut adalah hasil perhitungan matriks pengaruh total kriteria ternormalisasi.

Tabel 4.18 Matriks Pengaruh Total Kriteria Ternormalisasi

$D_u \backslash D_v$		D_1								D_2		D_3		
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
D_1	C_1	0.082	0.054	0.084	0.152	0.192	0.184	0.068	0.184	0.269	0.731	0.277	0.380	0.342
	C_2	0.179	0.051	0.040	0.200	0.176	0.164	0.022	0.168	0.313	0.687	0.185	0.280	0.535
	C_3	0.073	0.059	0.045	0.211	0.268	0.056	0.023	0.265	0.477	0.523	0.177	0.170	0.653
	C_4	0.161	0.141	0.086	0.117	0.201	0.072	0.026	0.196	0.393	0.607	0.271	0.280	0.449
	C_5	0.115	0.083	0.059	0.322	0.100	0.144	0.030	0.148	0.478	0.522	0.227	0.316	0.457
	C_6	0.259	0.066	0.051	0.278	0.149	0.082	0.031	0.084	0.255	0.745	0.388	0.281	0.331
	C_7	0.150	0.126	0.135	0.171	0.125	0.156	0.019	0.118	0.469	0.531	0.295	0.223	0.482
	C_8	0.094	0.076	0.057	0.478	0.101	0.065	0.034	0.095	0.441	0.559	0.289	0.292	0.419
D_2	C_9	0.174	0.144	0.057	0.254	0.100	0.152	0.033	0.085	0.316	0.684	0.279	0.274	0.447
	C_{10}	0.174	0.143	0.054	0.257	0.099	0.154	0.034	0.085	0.627	0.373	0.313	0.313	0.374
D_3	C_{11}	0.169	0.083	0.056	0.291	0.099	0.193	0.030	0.079	0.550	0.450	0.163	0.344	0.493
	C_{12}	0.217	0.108	0.048	0.234	0.172	0.118	0.027	0.076	0.560	0.440	0.280	0.178	0.542
	C_{13}	0.194	0.169	0.121	0.178	0.123	0.111	0.025	0.079	0.579	0.421	0.465	0.276	0.259

Berikut adalah hasil perhitungan matriks pengaruh total dimensi ternormalisasi.

Tabel 4.19 Matriks Pengaruh Total Dimensi Ternormalisasi

D_u	D_1	D_2	D_3
D_1	0.360	0.259	0.381
D_2	0.309	0.271	0.420
D_3	0.309	0.328	0.363

Lalu, dapat dilakukan pembuatan supermatriks tanpa bobot (*unweighted supermatrix*) berdasarkan matriks pengaruh total kriteria ternormalisasi. Berikut adalah hasil perancangan supermatriks tanpa bobot.

Tabel 4.20 Supermatriks Tanpa Bobot

$D_u \backslash D_v$		D_1								D_2		D_3		
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
D_1	C_1	0.082	0.179	0.073	0.161	0.115	0.259	0.150	0.094	0.174	0.174	0.169	0.217	0.194
	C_2	0.054	0.051	0.059	0.141	0.083	0.066	0.126	0.076	0.144	0.143	0.083	0.108	0.169
	C_3	0.084	0.040	0.045	0.086	0.059	0.051	0.135	0.057	0.057	0.054	0.056	0.048	0.121
	C_4	0.152	0.200	0.211	0.117	0.322	0.278	0.171	0.478	0.254	0.257	0.291	0.234	0.178
	C_5	0.192	0.176	0.268	0.201	0.100	0.149	0.125	0.101	0.100	0.099	0.099	0.172	0.123
	C_6	0.184	0.164	0.056	0.072	0.144	0.082	0.156	0.065	0.152	0.154	0.193	0.118	0.111
	C_7	0.068	0.022	0.023	0.026	0.030	0.031	0.019	0.034	0.033	0.034	0.030	0.027	0.025
	C_8	0.184	0.168	0.265	0.196	0.148	0.084	0.118	0.095	0.085	0.085	0.079	0.076	0.079
D_2	C_9	0.269	0.313	0.477	0.393	0.478	0.255	0.469	0.441	0.316	0.627	0.550	0.560	0.579
	C_{10}	0.731	0.687	0.523	0.607	0.522	0.745	0.531	0.559	0.684	0.373	0.450	0.440	0.421
D_3	C_{11}	0.277	0.185	0.177	0.271	0.227	0.388	0.295	0.289	0.279	0.313	0.163	0.280	0.465
	C_{12}	0.380	0.280	0.170	0.280	0.316	0.281	0.223	0.292	0.274	0.313	0.344	0.178	0.276
	C_{13}	0.342	0.535	0.653	0.449	0.457	0.331	0.482	0.419	0.447	0.374	0.493	0.542	0.259

Kemudian, dapat dilakukan perhitungan supermatriks berbobot (*weighted supermatrix*) berdasarkan transpose matriks pengaruh total dimensi ternormalisasi. Berikut adalah hasil perhitungan supermatriks berbobot.

Tabel 4.21 Supermatriks Berbobot

$D_u \backslash D_v$		D_1								D_2		D_3		
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
D_1	C_1	0.030	0.064	0.026	0.058	0.041	0.093	0.054	0.034	0.054	0.054	0.052	0.067	0.060
	C_2	0.020	0.018	0.021	0.051	0.030	0.024	0.045	0.027	0.045	0.044	0.026	0.033	0.052
	C_3	0.030	0.014	0.016	0.031	0.021	0.018	0.049	0.020	0.018	0.017	0.017	0.015	0.037
	C_4	0.054	0.072	0.076	0.042	0.116	0.100	0.062	0.172	0.078	0.079	0.090	0.072	0.055
	C_5	0.069	0.063	0.096	0.072	0.036	0.053	0.045	0.036	0.031	0.031	0.030	0.053	0.038
	C_6	0.066	0.059	0.020	0.026	0.052	0.030	0.056	0.023	0.047	0.048	0.060	0.036	0.034
	C_7	0.024	0.008	0.008	0.009	0.011	0.011	0.007	0.012	0.010	0.010	0.009	0.008	0.008
	C_8	0.066	0.061	0.095	0.070	0.053	0.030	0.042	0.034	0.026	0.026	0.025	0.024	0.024
D_2	C_9	0.070	0.081	0.124	0.102	0.124	0.066	0.122	0.114	0.085	0.170	0.181	0.184	0.190
	C_{10}	0.190	0.178	0.136	0.157	0.135	0.193	0.138	0.145	0.185	0.101	0.148	0.144	0.138
D_3	C_{11}	0.106	0.071	0.068	0.103	0.087	0.148	0.112	0.110	0.117	0.132	0.059	0.102	0.169
	C_{12}	0.145	0.107	0.065	0.107	0.120	0.107	0.085	0.111	0.115	0.132	0.125	0.064	0.100
	C_{13}	0.130	0.204	0.249	0.171	0.174	0.126	0.184	0.160	0.188	0.157	0.179	0.197	0.094
Σq_{ij}		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Selanjutnya, dapat dilakukan perhitungan limit supermatriks berbobot (*limiting supermatrix*). Berikut adalah hasil perhitungan limit supermatriks berbobot.

Tabel 4.22 Limit Supermatriks Berbobot

$D_u \backslash D_v$		D_1								D_2		D_3		
	$C_i \backslash C_j$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
D_1	C_1	0.0553	0.0553	0.0553	0.0553	0.0553	0.0553	0.0553	0.0553	0.0553	0.0553	0.0553	0.0553	0.0553
	C_2	0.0378	0.0378	0.0378	0.0378	0.0378	0.0378	0.0378	0.0378	0.0378	0.0378	0.0378	0.0378	0.0378
	C_3	0.0225	0.0225	0.0225	0.0225	0.0225	0.0225	0.0225	0.0225	0.0225	0.0225	0.0225	0.0225	0.0225
	C_4	0.0768	0.0768	0.0768	0.0768	0.0768	0.0768	0.0768	0.0768	0.0768	0.0768	0.0768	0.0768	0.0768
	C_5	0.0439	0.0439	0.0439	0.0439	0.0439	0.0439	0.0439	0.0439	0.0439	0.0439	0.0439	0.0439	0.0439
	C_6	0.0434	0.0434	0.0434	0.0434	0.0434	0.0434	0.0434	0.0434	0.0434	0.0434	0.0434	0.0434	0.0434
	C_7	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102	0.0102
	C_8	0.0357	0.0357	0.0357	0.0357	0.0357	0.0357	0.0357	0.0357	0.0357	0.0357	0.0357	0.0357	0.0357
D_2	C_9	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400
	C_{10}	0.1492	0.1492	0.1492	0.1492	0.1492	0.1492	0.1492	0.1492	0.1492	0.1492	0.1492	0.1492	0.1492
D_3	C_{11}	0.1144	0.1144	0.1144	0.1144	0.1144	0.1144	0.1144	0.1144	0.1144	0.1144	0.1144	0.1144	0.1144
	C_{12}	0.1097	0.1097	0.1097	0.1097	0.1097	0.1097	0.1097	0.1097	0.1097	0.1097	0.1097	0.1097	0.1097
	C_{13}	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612

Selanjutnya, didapatkan bobot lokal kriteria dari supermatriks tanpa bobot (*unweighted supermatrix*) serta bobot lokal dimensi dari transpose matriks pengaruh total (*total-influential matrix*) dimensi ternormalisasi. Didapatkan pula bobot global kriteria dari limit supermatriks berbobot (*weighted supermatrix*).

Tabel 4.23 Bobot Lokal dan Global Kriteria

D_u	C_i	w_u^D	w_i^C	Prioritas Dimensi	Prioritas Kriteria	w_i	Prioritas Global
D_1	C_1	0.326	0.157	2	2	0.055	7
	C_2		0.100		6	0.038	10
	C_3		0.069		7	0.022	12
	C_4		0.242		1	0.077	6
	C_5		0.146		3	0.044	8
	C_6		0.127		5	0.043	9
	C_7		0.031		8	0.010	13
	C_8		0.128		4	0.036	11
D_2	C_9	0.286	0.441	3	2	0.140	3
	C_{10}		0.559		1	0.149	2
D_3	C_{11}	0.388	0.278	1	2	0.114	4
	C_{12}		0.277		3	0.110	5
	C_{13}		0.445		1	0.161	1

Dari tabel di atas, dapat diketahui bahwa urutan prioritas dimensi dari yang memiliki prioritas paling tinggi ke rendah adalah: dimensi lingkungan, dimensi ekonomi, dan dimensi sosial. Urutan prioritas kriteria dalam dimensi lingkungan dari yang paling tinggi ke rendah adalah: kriteria konsumsi sumber daya, kriteria sistem manajemen lingkungan, dan kriteria desain produk ramah lingkungan. Urutan prioritas kriteria dalam dimensi ekonomi dari yang paling tinggi ke rendah adalah: kontinuitas, harga, waktu tunggu, responsivitas, kualitas, produktivitas, kapasitas, dan teknologi produksi. Sementara urutan prioritas kriteria dalam dimensi sosial dari yang paling tinggi ke rendah adalah: aktivitas pendukung serta sistem manajemen kesehatan dan keselamatan kerja.

Berikut adalah hasil matriks keputusan (*decision matrix*) mengenai penilaian kinerja pemasok yang didasarkan pada penelitian Goren (2018).

Tabel 4.24 Matriks Penilaian Kinerja Pemasok

$S_g \backslash C_i$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
S_1	1%	100%	63%	99%	1%	52%	65%	7%	76%	50%	80%	95%	10%
S_2	6%	75%	97%	82%	1%	93%	82%	5%	92%	54%	54%	59%	1%
S_3	9%	86%	86%	69%	3%	56%	79%	8%	83%	87%	62%	52%	6%
S_4	10%	63%	87%	97%	7%	55%	50%	10%	97%	60%	82%	79%	5%
S_5	6%	75%	57%	62%	10%	91%	54%	2%	50%	76%	91%	65%	2%
S_6	7%	98%	82%	51%	9%	66%	66%	2%	79%	78%	51%	97%	8%

Berikut adalah hasil penentuan kategori kriteria dan koefisien kerugian yang didasarkan pada penelitian Goren (2018).

Tabel 4.25 Kategori Kriteria dan Koefisien Kerugian

C_i	w_i	Kategori	Objektif	Karakteristik Kerugian	Nilai yang Diinginkan	Rentang yang Diterima	Y_i/X_i	$L(Y_i)/L(X_i)$	k_i
C_1	0.055	<i>Risk</i>	<i>Min</i>	<i>Lower the better</i>	0%	0-90	0.9	100	123.46
C_2	0.038	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	65-100	0.65	100	42.25
C_3	0.022	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	65-100	0.65	100	42.25
C_4	0.077	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	75-100	0.75	100	56.25
C_5	0.044	<i>Risk</i>	<i>Min</i>	<i>Lower the better</i>	0%	0-10	0.1	100	10000.00
C_6	0.043	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	70-100	0.7	100	49.00
C_7	0.010	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	60-100	0.6	100	36.00
C_8	0.036	<i>Risk</i>	<i>Min</i>	<i>Lower the better</i>	0%	0-10	0.1	100	10000.00
C_9	0.140	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	75-100	0.75	100	56.25
C_{10}	0.149	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	75-100	0.75	100	56.25
C_{11}	0.114	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	80-100	0.8	100	64.00
C_{12}	0.110	<i>Benefit</i>	<i>Max</i>	<i>Higher the better</i>	100%	80-100	0.8	100	64.00
C_{13}	0.161	<i>Risk</i>	<i>Min</i>	<i>Lower the better</i>	0%	0-10	0.1	100	10000.00

Dengan menggunakan penilaian kinerja pemasok dan koefisien kerugian, dapat dilakukan penentuan skor kerugian (*loss score*). Berikut adalah hasil perhitungan dari penentuan skor kerugian pada setiap kriteria.

Tabel 4.26 *Loss Score* Pemasok Pada Setiap Kriteria

$S_g \setminus C_i$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
S_1	0.0123	42.250	106.450	57.392	1.00	181.213	85.207	49.00	97.386	225.000	100.000	70.914	100.00
S_2	0.4444	75.111	44.904	83.656	1.00	56.654	53.540	25.00	66.458	192.901	219.479	183.855	1.00
S_3	1.0000	57.125	57.125	118.147	9.00	156.250	57.683	64.00	81.652	74.316	166.493	236.686	36.00
S_4	1.2346	106.450	55.820	59.783	49.00	161.983	144.000	100.00	59.783	156.250	95.181	102.548	25.00
S_5	0.4444	75.111	130.040	146.332	100.00	59.172	123.457	4.00	225.000	97.386	77.285	151.479	4.00
S_6	0.6049	43.992	62.835	216.263	81.00	112.489	82.645	4.00	90.130	92.456	246.059	68.020	64.00

Selanjutnya, dapat dilakukan perhitungan skor kerugian total (*total loss score*) dan *rating* pemasok berdasarkan skor kerugian dan bobot global. Berikut adalah hasil perhitungan dari skor kerugian total dan *rating* pemasok.

Tabel 4.27 *Total Loss Score* dan *Rating* Pemasok

$S_g \setminus C_i$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	TL_g	W_g	<i>Rank</i>
S_1	0.001	1.597	2.391	4.41	0.044	7.860	0.872	1.750	13.63	33.58	11.44	7.78	16.12	101.47	0.1569	5
S_2	0.025	2.839	1.009	6.42	0.044	2.457	0.548	0.893	9.30	28.79	25.10	20.17	0.16	97.75	0.1629	4
S_3	0.055	2.159	1.283	9.07	0.395	6.777	0.590	2.286	11.43	11.09	19.04	25.96	5.80	95.94	0.1659	2
S_4	0.068	4.024	1.254	4.59	2.151	7.026	1.473	3.572	8.37	23.32	10.88	11.25	4.03	82.01	0.1941	1
S_5	0.025	2.839	2.921	11.24	4.389	2.567	1.263	0.143	31.50	14.53	8.84	16.61	0.64	97.51	0.1633	3
S_6	0.033	1.663	1.412	16.61	3.555	4.879	0.846	0.143	12.62	13.80	28.14	7.46	10.32	101.47	0.1569	6

Rating pemasok ini selanjutnya digunakan dalam model optimasi pemilihan pemasok dan alokasi pesanan sebagai parameter W_i yang merupakan *rating* untuk pemasok ke- i .

4.4.2 Data Nilai Parameter

Data nilai parameter yang digunakan dalam contoh numerik model optimasi pemilihan pemasok dan alokasi pesanan didasarkan pada penelitian Goren (2018) dan Hosseini, dkk. (2019) dengan penyesuaian pada parameter baru. Dalam contoh numerik ini, satuan biaya diekspresikan dalam USD (\$). Berikut adalah nilai himpunan indeks dan parameter yang digunakan dalam contoh numerik.

Tabel 4.28 Nilai dari Indeks dan Parameter

Notasi	Nilai	Notasi	Nilai
1. Indeks			
I	$\{1,2,3, \dots, 6\}$	T	$\{1,2,3, \dots, 10\}$
R	$\{1,2,3, \dots, 5\}$	S	$\{1,2,3, \dots, 5\}$
2. Parameter			
α	0.9	CU	uniform (0.03, 0.08)
ε	1	CD	uniform (1, 2)
δ	1	P^s	$\{s \mid \text{uniform}(0.1, 0.85), \forall s \in S\}$
LL	0.05	C_i	$\{i \mid \text{uniform}(160, 200), \forall i \in I\}$
SD	600	λ_i	$\{i \mid \text{uniform}(0.3, 0.8), \forall i \in I\}$
v	25	f_i	$\{i \mid \text{uniform}(7500, 8500), \forall i \in I\}$
Δ	35000	B_i	$\{i \mid \text{uniform}(1100, 1250), \forall i \in I\}$
Ψ	10	G_i	$\{i \mid \text{uniform}(5800, 6900), \forall i \in I\}$
ξ	0.5	Φ_i	$\{i \mid \text{uniform}(1000, 2500), \forall i \in I\}$
φ	0.5	c_i	$\{i \mid \text{randint}(1, 3), \forall i \in I\}$
RC	5	η_i	$\{i \mid \text{uniform}(300, 700), \forall i \in I\}$
RL	3	BI_r	$\{r \mid \text{randint}(0, 20), \forall r \in R\}$

Berikut adalah nilai dari parameter terkait dengan pemasok yang digunakan dalam contoh numerik.

Tabel 4.29 Nilai dari Parameter terkait Pemasok

Pemasok	W_i	E_i	μ_i	A_i
$i = 1$	0.1569	randint (15, 25)	uniform (100, 150)	randint (4, 5)
$i = 2$	0.1629	randint (20, 30)	uniform (100, 120)	randint (2, 4)
$i = 3$	0.1659	randint (12,25)	uniform (120, 150)	randint (1, 3)
$i = 4$	0.1941	randint (15,25)	uniform (150, 180)	randint (1, 4)
$i = 5$	0.1633	randint (20,30)	uniform (100, 120)	randint (3, 5)
$i = 6$	0.1569	randint (18,30)	uniform (150, 180)	randint (2, 5)

Tabel 4.29 Nilai dari Parameter terkait Pemasok (lanjutan)

Pemasok	g_i	θ_i	AT_i	TT_i
$i = 1$	800	uniform (0.03, 0.05)	uniform (720, 800)	uniform (0.62, 0.70)
$i = 2$	780	uniform (0.02, 0.07)	uniform (650, 720)	uniform (0.59, 0.67)
$i = 3$	950	uniform (0.04, 0.07)	uniform (760, 830)	uniform (0.58, 0.60)
$i = 4$	700	uniform (0.05, 0.07)	uniform (680, 750)	uniform (0.65, 0.72)
$i = 5$	820	uniform (0.02, 0.08)	uniform (650, 750)	uniform (0.62, 0.71)
$i = 6$	650	uniform (0.03, 0.05)	uniform (700, 780)	uniform (0.65, 0.70)

Berikut adalah nilai dari parameter jarak antar pemasok yang digunakan dalam contoh numerik.

Tabel 4.30 Nilai dari Parameter Jarak antar Pemasok

d_{ij}	j					
i	1	2	3	4	5	6
1	0	480	550	680	925	750
2	480	0	450	740	650	560
3	550	450	0	695	870	685
4	680	740	695	0	680	490
5	925	650	870	680	0	875
6	750	560	685	490	875	0

Berikut adalah nilai dari parameter terkait dengan pemasok yang digunakan dalam contoh numerik.

Tabel 4.31 Nilai dari Parameter terkait Produk

Produk	H_r	LO_r	Q_r	IC_r	LT_r
$r = 1$	uniform (1, 3)	uniform (12, 18)	0.06	1500	10080
$r = 2$	uniform (3, 6)	uniform (18, 22)	0.075	1000	11520
$r = 3$	uniform (2, 5)	uniform (26, 30)	0.06	1250	10800
$r = 4$	uniform (1, 3)	uniform (18, 22)	0.08	1250	12240
$r = 5$	uniform (2, 5)	uniform (26, 30)	0.075	1000	10080

Berikut adalah nilai yang digunakan dalam contoh numerik untuk parameter permintaan pada setiap produk.

Tabel 4.32 Nilai dari Parameter Permintaan

Produk (r)	Permintaan (D_{rt})
1	$\{t \mid \text{randint}(470, 500), \forall t \in T\}$
2	$\{t \mid \text{randint}(160, 190), \forall t \in T\}$
3	$\{t \mid \text{randint}(380, 400), \forall t \in T\}$
4	$\{t \mid \text{randint}(350, 380), \forall t \in T\}$
5	$\{t \mid \text{randint}(200, 230), \forall t \in T\}$

Berikut adalah nilai yang digunakan dalam contoh numerik untuk parameter fraksi kapasitas yang bertahan dalam skenario disrupsi.

Tabel 4.33 Nilai dari Parameter Fraksi Kapasitas Bertahan

Pemasok	Fraksi Kapasitas Bertahan (τ_i^s)
$i = 1$	$\{s \mid \text{uniform } (0.2, 0.4), \forall s \in S\}$
$i = 2$	$\{s \mid \text{uniform } (0.4, 0.6), \forall s \in S\}$
$i = 3$	$\{s \mid \text{uniform } (0.2, 0.6), \forall s \in S\}$
$i = 4$	$\{s \mid \text{uniform } (0.3, 0.8), \forall s \in S\}$
$i = 5$	$\{s \mid \text{uniform } (0.2, 0.5), \forall s \in S\}$
$i = 6$	$\{s \mid \text{uniform } (0.4, 0.7), \forall s \in S\}$

Berikut adalah nilai yang digunakan dalam contoh numerik untuk parameter harga unit produk.

Tabel 4.34 Nilai dari Parameter Harga Produk

K_{ir}	r				
i	1	2	3	4	5
1	uniform (11, 15)	uniform (17, 18)	uniform (25, 27)	uniform (18, 20)	uniform (25, 28)
2	uniform (10, 12)	uniform (18, 20)	uniform (24, 26)	uniform (17, 19)	uniform (26, 29)
3	uniform (12, 13)	uniform (17, 19)	uniform (25, 27)	uniform (15, 17)	uniform (25, 27)
4	uniform (13, 15)	uniform (18, 20)	uniform (23, 25)	uniform (15, 18)	uniform (28, 31)
5	uniform (10, 13)	uniform (17, 19)	uniform (24, 25)	uniform (20, 21)	uniform (29, 32)
6	uniform (11, 13)	uniform (18, 19)	uniform (25, 26)	uniform (17, 18)	uniform (24, 27)

Berikut adalah nilai yang digunakan dalam contoh numerik untuk parameter biaya pesan.

Tabel 4.35 Nilai dari Parameter Biaya Pesan

O_{ir}	r				
i	1	2	3	4	5
1	uniform (120, 150)	uniform (130, 160)	uniform (100, 120)	uniform (140, 160)	uniform (80, 100)
2	uniform (90, 130)	uniform (110, 150)	uniform (80, 110)	uniform (140, 170)	uniform (100, 120)
3	uniform (120, 160)	uniform (90, 120)	uniform (90, 100)	uniform (120, 150)	uniform (130, 140)
4	uniform (100, 150)	uniform (150, 170)	uniform (90, 120)	uniform (100, 160)	uniform (110, 150)
5	uniform (80, 130)	uniform (130, 160)	uniform (120, 140)	uniform (140, 160)	uniform (110, 140)
6	uniform (100, 130)	uniform (90, 130)	uniform (80, 130)	uniform (120, 160)	uniform (80, 120)

Berikut adalah nilai yang digunakan dalam contoh numerik untuk parameter waktu *set-up*.

Tabel 4.36 Nilai dari Parameter Waktu *Set-up*

ST_{ir}	r				
i	1	2	3	4	5
1	uniform (10, 15)	uniform (5, 10)	uniform (10, 12)	uniform (15, 20)	uniform (10, 15)
2	uniform (15, 20)	uniform (4, 10)	uniform (12, 15)	uniform (20, 22)	uniform (15, 20)
3	uniform (10, 15)	uniform (5, 10)	uniform (8, 12)	uniform (15, 22)	uniform (10, 15)
4	uniform (20, 25)	uniform (4, 8)	uniform (10, 15)	uniform (15, 20)	uniform (15, 20)
5	uniform (15, 20)	uniform (6, 12)	uniform (8, 12)	uniform (20, 23)	uniform (15, 20)
6	uniform (20, 25)	uniform (5, 9)	uniform (10, 12)	uniform (20, 22)	uniform (10, 15)

Berikut adalah nilai yang digunakan dalam contoh numerik untuk parameter waktu produksi.

Tabel 4.37 Nilai dari Parameter Waktu Produksi

PT_{ir}	r				
i	1	2	3	4	5
1	uniform (10, 15)	uniform (9, 13)	uniform (12, 15)	uniform (7, 10)	uniform (8, 11)
2	uniform (10, 16)	uniform (8, 12)	uniform (12, 17)	uniform (9, 10)	uniform (6, 12)
3	uniform (12, 15)	uniform (8, 14)	uniform (10, 16)	uniform (8, 12)	uniform (7, 12)
4	uniform (11, 16)	uniform (10, 15)	uniform (11, 15)	uniform (7, 12)	uniform (8, 10)
5	uniform (10, 15)	uniform (10, 12)	uniform (11, 14)	uniform (7, 10)	uniform (8, 11)
6	uniform (11, 16)	uniform (9, 12)	uniform (10, 17)	uniform (9, 11)	uniform (6, 10)

4.4.3 Pengolahan Pemilihan Pemasok dan Alokasi Pesanan

Hasil pengolahan pemilihan pemasok dan alokasi pesanan didapatkan dari eksekusi kode pemrograman MO-MILP pada Python 3. Program dijalankan selama total 160 jam dan menghasilkan solusi optimum lokal dengan *MIP Gap* sebesar 0.3%. Berikut adalah hasil solusi optimal untuk variabel kuantitas pesanan (X_{irt}^s) dan keputusan pesan (Y_{irt}^s) pada pemasok ke- i untuk produk r dalam periode t dan skenario disrupsi s .

Tabel 4.38 Solusi Optimal untuk Kuantitas Pesanan dan Keputusan Pesan

<i>r</i>		1																			
Var.		X_{irt}^s										Y_{irt}^s									
<i>s</i>	<i>i \ t</i>	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	125	129	0	102	114	0	0	0	0	0	1	1	0	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	44	0	0	40	0	0	0	0	0	0	1	0	0	1	0	0
	6	466	493	561	611	611	611	611	611	610	609	1	1	1	1	1	1	1	1	1	1
2	1	102	0	0	0	0	120	122	0	102	114	1	0	0	0	0	1	1	0	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	39	53	0	3	40	0	0	0	0	0	1	1	0	1	1	0	0
	6	341	493	538	611	595	611	611	611	610	609	1	1	1	1	1	1	1	1	1	1
3	1	0	0	0	0	97	0	128	0	102	112	0	0	0	0	1	0	1	0	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	57	0	40	0	0	0	0	0	0	0	1	0	1	0	0
	6	466	493	578	611	611	611	611	611	610	610	1	1	1	1	1	1	1	1	1	1
4	1	0	0	0	86	0	0	129	0	0	111	0	0	0	1	0	0	1	0	0	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	58	57	0	40	46	0	0	0	0	0	1	1	0	1	1	0
	6	466	493	611	611	586	611	611	611	610	611	1	1	1	1	1	1	1	1	1	1
5	1	89	0	0	86	0	0	129	0	100	114	1	0	0	1	0	0	1	0	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	53	57	0	40	0	0	0	0	0	0	1	1	0	1	0	0
	6	354	493	551	611	595	611	611	611	611	609	1	1	1	1	1	1	1	1	1	1

Tabel 4.38 Solusi Optimal untuk Kuantitas Pesanan dan Keputusan Pesan (lanjutan)

<i>r</i>		2																			
Var.		X_{irt}^s										Y_{irt}^s									
<i>s</i>	<i>i \ t</i>	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	1	0	181	207	0	0	154	180	363	256	243	0	1	1	0	0	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	191	167	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	6	174	0	0	0	0	182	177	41	119	134	1	0	0	0	0	1	1	1	1	1
2	1	0	151	181	76	0	130	158	333	242	243	0	1	1	1	0	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	39	0	160	167	0	0	0	0	0	0	1	0	1	1	0	0	0	0
	6	157	0	0	0	0	201	195	64	130	134	1	0	0	0	0	1	1	1	1
3	1	151	181	193	0	0	303	181	363	257	245	1	1	1	0	0	1	1	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	176	167	0	0	0	0	0	0	0	0	1	1	0	0	0	0
	6	23	0	0	0	0	63	174	41	118	133	1	0	0	0	0	1	1	1	1
4	1	120	151	171	0	0	272	149	333	363	247	1	1	1	0	0	1	1	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	194	167	0	0	0	0	0	0	0	0	1	1	0	0	0	0
	6	84	0	0	0	0	88	202	64	33	131	1	0	0	0	0	1	1	1	1
5	1	0	0	0	0	0	257	134	318	229	243	0	0	0	0	0	1	1	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	147	169	0	167	0	0	0	0	0	0	1	1	0	1	0	0	0	0
	6	157	43	0	233	0	100	214	76	140	134	1	1	0	1	0	1	1	1	1

Tabel 4.38 Solusi Optimal untuk Kuantitas Pesanan dan Keputusan Pesan (lanjutan)

r		3																			
Var.		X_{irt}^s										Y_{irt}^s									
s	i/t	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	1	117	0	0	188	211	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	109	141	172	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	6	151	244	217	400	401	554	550	573	557	569	1	1	1	1	1	1	1	1	1	1
2	1	0	0	0	103	188	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	114	188	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
	6	358	271	201	490	419	554	550	573	557	569	1	1	1	1	1	1	1	1	1	1
3	1	0	0	0	188	122	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	93	125	156	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	6	284	260	233	418	472	554	550	573	557	569	1	1	1	1	1	1	1	1	1	1
4	1	0	0	0	84	188	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5	5	125	156	188	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
	6	252	229	205	505	419	554	550	573	557	569	1	1	1	1	1	1	1	1	1
	1	0	105	0	0	176	0	0	0	0	0	0	1	0	0	1	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	220	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	6	358	300	557	187	429	554	550	573	557	569	1	1	1	1	1	1	1	1	1

Tabel 4.38 Solusi Optimal untuk Kuantitas Pesanan dan Keputusan Pesan (lanjutan)

r		4																			
Var.		X_{irt}^s										Y_{irt}^s									
s	$i \backslash t$	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	380	387	0	0	0	0	0	0	0	0	1	1	0	0	0
	6	364	370	378	365	558	0	0	672	718	686	1	1	1	1	1	0	0	1	1	1
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	177	0	0	0	0	385	387	0	0	0	1	0	0	0	0	1	1	0	0	0
	6	168	370	378	365	553	0	0	672	718	686	1	1	1	1	1	0	0	1	1	1
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	57	255	384	0	0	0	0	0	0	0	1	1	1	0	0	0
	6	364	370	378	365	402	227	0	672	718	686	1	1	1	1	1	1	0	1	1	1
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	300	387	0	0	0	0	0	0	0	0	1	1	0	0	0
	6	364	370	378	365	490	148	0	672	718	686	1	1	1	1	1	1	0	1	1	1
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	177	0	0	0	0	300	387	0	0	0	1	0	0	0	0	1	1	0	0	0
	6	184	370	378	365	490	148	0	672	718	686	1	1	1	1	1	1	0	1	1	1

Tabel 4.38 Solusi Optimal untuk Kuantitas Pesanan dan Keputusan Pesan (lanjutan)

r		5																			
Var.		X_{irt}^s										Y_{irt}^s									
s	$i \backslash t$	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	212	217	230	226	363	394	410	365	390	362	1	1	1	1	1	1	1	1	1	1
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	209	217	230	226	363	394	410	365	390	362	1	1	1	1	1	1	1	1	1	1
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	212	217	230	226	363	394	410	365	390	362	1	1	1	1	1	1	1	1	1	1
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	212	217	230	226	363	394	410	365	390	362	1	1	1	1	1	1	1	1	1	1
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	212	217	230	226	363	394	410	365	390	362	1	1	1	1	1	1	1	1	1	1

Berikut adalah hasil solusi optimal untuk variabel kuantitas kehilangan penjualan (L_{rt}^s) dan tingkat persediaan (I_{rt}^s) untuk produk r dalam periode t dan skenario s .

Tabel 4.39 Solusi Optimal untuk Kuantitas Kehilangan Penjualan dan Tingkat Persediaan

Var.		L_{rt}^s										I_{rt}^s									
r	$s \backslash t$	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0	0	0	0	65	202	377	617	859	1035	1265	1500
	2	23	0	0	0	0	0	0	0	0	0	0	0	42	218	386	621	859	1035	1265	1500
	3	0	0	0	0	0	0	0	0	0	0	0	0	82	219	447	619	860	1036	1266	1500
	4	0	0	0	0	0	0	0	0	0	0	0	0	115	338	502	674	916	1092	1266	1500
	5	23	0	0	0	0	0	0	0	0	0	0	0	55	278	446	618	860	1036	1265	1500

2	1	0	0	0	0	0	0	0	0	0	0	0	9	0	42	43	47	213	391	611	806	1000
	2	8	0	0	0	0	0	0	0	0	0	0	0	0	16	62	66	227	401	614	806	1000
	3	0	0	0	0	0	0	0	0	0	0	0	9	0	28	14	18	214	390	610	805	1000
	4	0	0	0	0	0	0	0	0	0	0	0	39	0	6	10	14	204	376	589	805	1000
	5	8	0	0	0	0	0	0	0	0	0	0	0	0	4	47	51	238	407	617	806	1000
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	189	404	571	737	910	1078	1250
	2	19	0	0	0	0	0	0	0	0	0	0	0	0	0	194	404	571	737	910	1078	1250
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	207	404	571	737	910	1078	1250
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	4	194	404	571	737	910	1078	1250
	5	19	0	0	0	0	0	0	0	0	0	0	0	20	188	196	404	571	737	910	1078	1250
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	228	261	581	923	1250
	2	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	195	228	261	581	923	1250
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	101	231	261	581	923	1250
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	132	228	261	581	923	1250
	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	132	228	261	581	923	1250
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	326	507	668	840	1000
	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	326	507	668	840	1000
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	326	507	668	840	1000
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	326	507	668	840	1000
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	326	507	668	840	1000

Berikut adalah hasil solusi optimal untuk variabel kapasitas kumulatif (QC_{it}^s) dan kapasitas yang terpulihkan (k_{it}^s) pada pemasok ke- i dalam periode t dan skenario s .

Tabel 4.40 Solusi Optimal untuk Kapasitas Kumulatif dan Kapasitas yang Terpulihkan

Vr.		QC_{it}^s										k_{it}^s									
i	$s \backslash t$	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	1	10	12	14	16	18	20	22	24	25	25	2	2	2	2	2	2	2	1	0	0
	2	8	10	12	14	16	18	20	22	24	25	2	2	2	2	2	2	2	2	1	0
	3	10	12	14	16	18	20	22	24	25	25	2	2	2	2	2	2	2	1	0	0
	4	8	10	12	14	16	18	20	22	24	25	2	2	2	2	2	2	2	2	1	0
	5	7	9	11	13	15	17	19	21	23	25	2	2	2	2	2	2	2	2	2	0
2	1	13	15	17	19	21	23	24	24	24	24	2	2	2	2	2	1	0	0	0	0
	2	15	17	19	21	23	24	24	24	24	24	2	2	2	2	1	0	0	0	0	0
	3	13	15	17	19	21	23	24	24	24	24	2	2	2	2	2	1	0	0	0	0
	4	10	12	14	16	18	20	22	24	24	24	2	2	2	2	2	2	2	0	0	0
	5	11	13	15	17	19	21	23	24	24	24	2	2	2	2	2	2	1	0	0	0
3	1	5	7	9	11	13	15	17	18	18	18	2	2	2	2	2	2	1	0	0	0
	2	5	7	9	11	13	15	17	18	18	18	2	2	2	2	2	2	1	0	0	0
	3	9	11	13	15	17	18	18	18	18	18	2	2	2	2	1	0	0	0	0	0
	4	10	12	14	16	18	18	18	18	18	18	2	2	2	2	0	0	0	0	0	0
	5	6	8	10	12	14	16	18	18	18	18	2	2	2	2	2	2	0	0	0	0
4	1	11	13	15	17	19	21	23	23	23	23	2	2	2	2	2	2	0	0	0	0
	2	8	10	12	14	16	18	20	22	23	23	2	2	2	2	2	2	2	1	0	0
	3	18	20	22	23	23	23	23	23	23	23	2	2	1	0	0	0	0	0	0	0
	4	18	20	22	23	23	23	23	23	23	23	2	2	1	0	0	0	0	0	0	0
	5	17	19	21	23	23	23	23	23	23	23	2	2	2	0	0	0	0	0	0	0

5	1	7	9	11	13	15	17	19	21	22	22	2	2	2	2	2	2	2	1	0	0
	2	8	10	12	14	16	18	20	22	22	22	2	2	2	2	2	2	2	0	0	0
	3	6	8	10	12	14	16	18	20	22	22	2	2	2	2	2	2	2	2	0	0
	4	8	10	12	14	16	18	20	22	22	22	2	2	2	2	2	2	2	0	0	0
	5	8	10	12	14	16	18	20	22	22	22	2	2	2	2	2	2	2	0	0	0
6	1	95	111	127	143	159	175	188	188	188	188	16	16	16	16	16	13	0	0	0	0
	2	77	93	109	125	141	157	173	188	188	188	16	16	16	16	16	16	15	0	0	0
	3	111	127	143	159	175	188	188	188	188	188	16	16	16	16	13	0	0	0	0	0
	4	119	135	151	167	183	188	188	188	188	188	16	16	16	16	5	0	0	0	0	0
	5	79	95	111	127	143	159	175	188	188	188	16	16	16	16	16	16	13	0	0	0

Berikut adalah hasil solusi optimal untuk variabel pemilihan pemasok (Z_i), kapasitas restoratif tambahan (U_i), variabel *auxiliary* terkait dengan jumlah disrupsi (w_i), kapasitas produksi tambahan (ω_i), dan jumlah disrupsi (N_i) pada pemasok ke- i serta biaya penalty ketahanan (Ω^s) dalam skenario s .

Tabel 4.41 Solusi Optimal untuk Variabel terkait Pemasok dan Skenario Disrupsi

i	Z_i	U_i	N_i	w_i	ω_i	s	Ω_s
1	1	0	6	6	1	1	265858
2	0	0	12	0	0	2	283196
3	0	2	8	0	0	3	240007
4	0	5	6	0	0	4	238280
5	1	0	6	6	2	5	270599
6	1	11	7	7	165		

Berikut adalah hasil solusi optimal variabel pemilihan pasangan pemasok (r_{ij}) untuk pemasok ke- i dan j .

Tabel 4.42 Solusi Optimal untuk Pemilihan Pasangan Pemasok

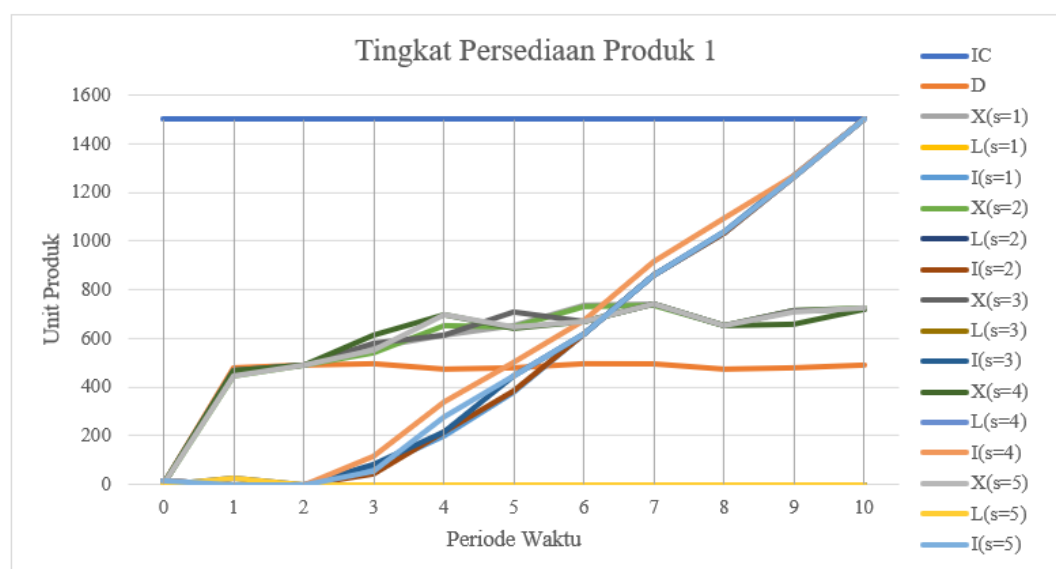
r_{ij}	j					
i	1	2	3	4	5	6
1		0	0	0	1	1
2			0	0	0	0
3				0	0	0
4					0	0
5						1
6						

Berikut adalah hasil nilai optimal dari fungsi objektif total biaya gabungan atau *Joint Total Cost (JTC)*, total nilai pembelian atau *Total Value Purchasing (TVP)*, dan segregasi geografis atau *Geographical Segregation (GS)*.

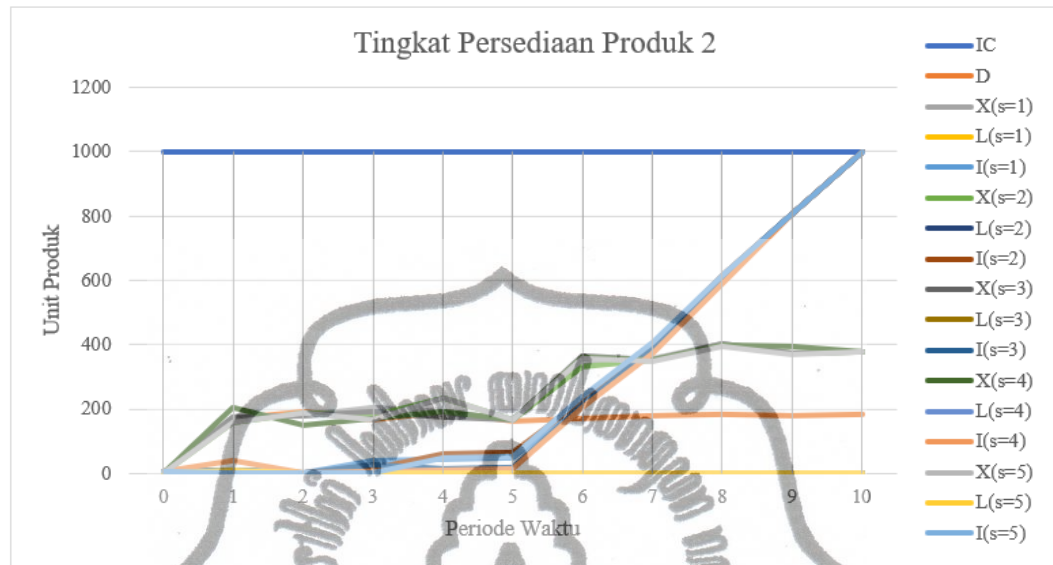
Tabel 4.43 Nilai Optimal dari Fungsi Objektif

Objektif	Nilai	Satuan	Persentase
<i>PC</i>	961945.802	\$	17.495%
<i>HC</i>	129804.814	\$	2.361%
<i>OC</i>	18672.580	\$	0.340%
<i>SCC</i>	3511.807	\$	0.064%
<i>TRC</i>	11475.933	\$	0.209%
<i>LSP</i>	546.203	\$	0.010%
<i>MTC</i>	1125957.139	\$	20.477%
<i>ACC</i>	1300805.190	\$	23.657%
<i>RCI</i>	114752.167	\$	2.087%
<i>DC</i>	29736.027	\$	0.541%
<i>SRC</i>	286348.373	\$	5.208%
<i>SOC</i>	2006196.817	\$	36.486%
<i>PDP</i>	54624.166	\$	0.993%
<i>STC</i>	3792462.740	\$	68.972%
<i>MC</i>	290059.722	\$	5.275%
<i>CC</i>	290059.722	\$	5.275%
<i>JTC</i>	5498539.324	\$	
<i>TVP</i>	7962.473	Unit Produk	
<i>GS</i>	2550	km	

Berikut adalah grafik tingkat persediaan untuk produk $r = 1$ dari solusi optimal yang didapatkan.

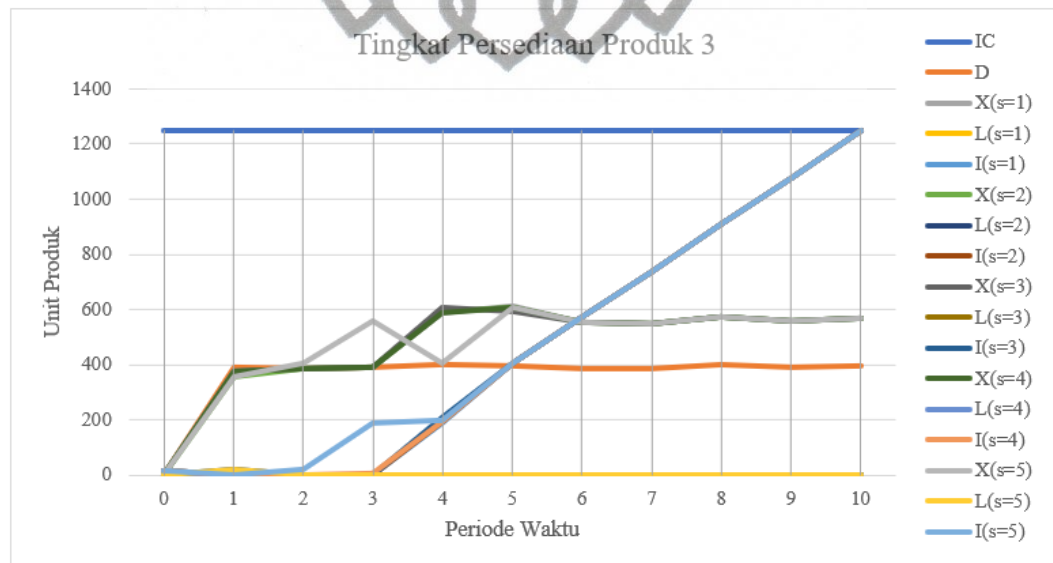
**Gambar 4.11** Tingkat Persediaan untuk Produk 1

Berikut adalah grafik tingkat persediaan untuk produk $r = 2$ dari solusi optimal yang didapatkan.



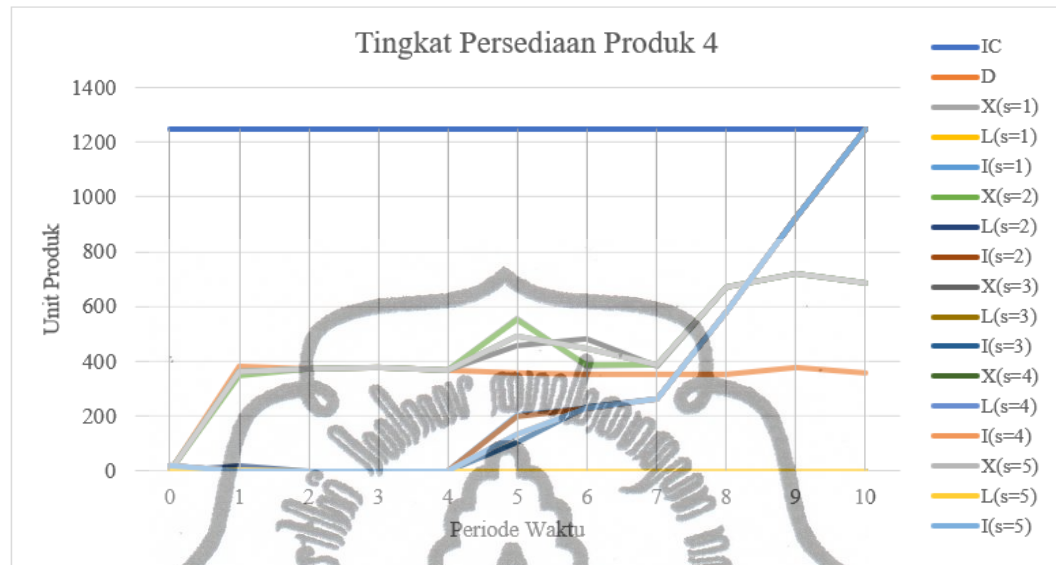
Gambar 4.12 Tingkat Persediaan untuk Produk 2

Berikut adalah grafik tingkat persediaan untuk produk $r = 3$ dari solusi optimal yang didapatkan.



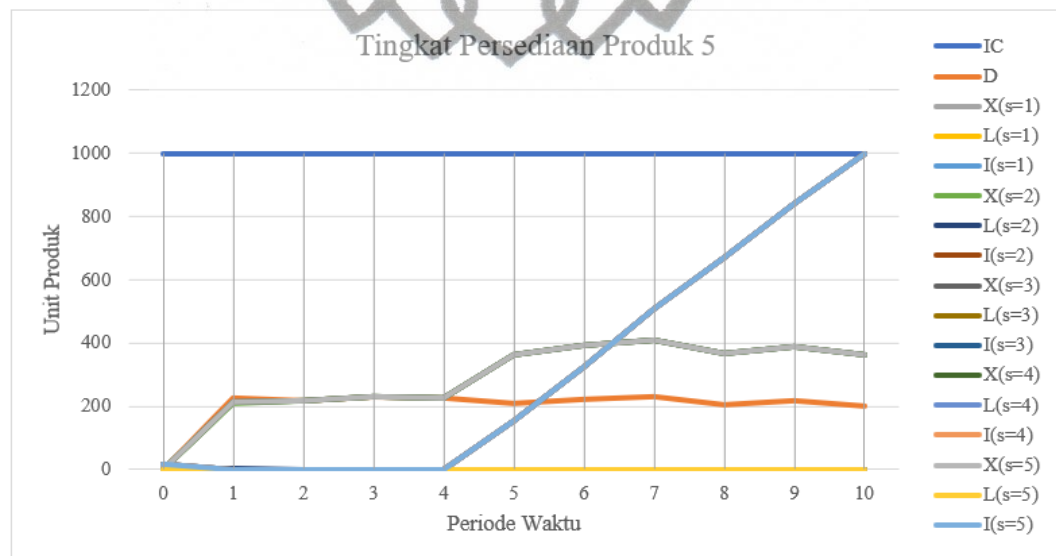
Gambar 4.13 Tingkat Persediaan untuk Produk 3

Berikut adalah grafik tingkat persediaan untuk produk $r = 4$ dari solusi optimal yang didapatkan.



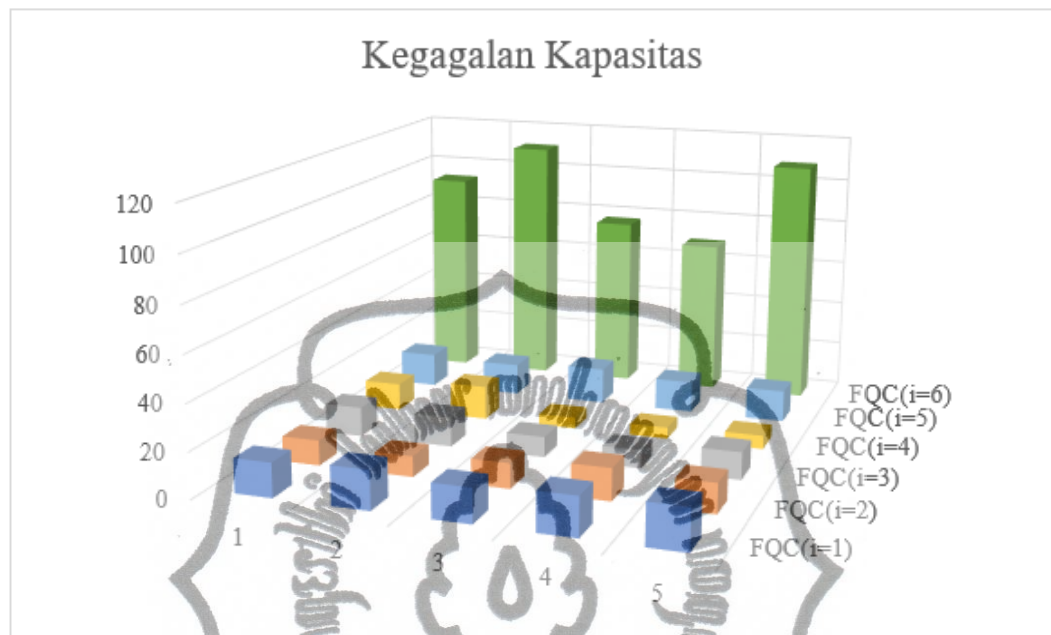
Gambar 4.14 Tingkat Persediaan untuk Produk 4

Berikut adalah grafik tingkat persediaan untuk produk $r = 5$ dari solusi optimal yang didapatkan.



Gambar 4.15 Tingkat Persediaan untuk Produk 5

Berikut adalah grafik jumlah kegagalan kapasitas pada setiap skenario disrupsi dari solusi optimal yang didapatkan.



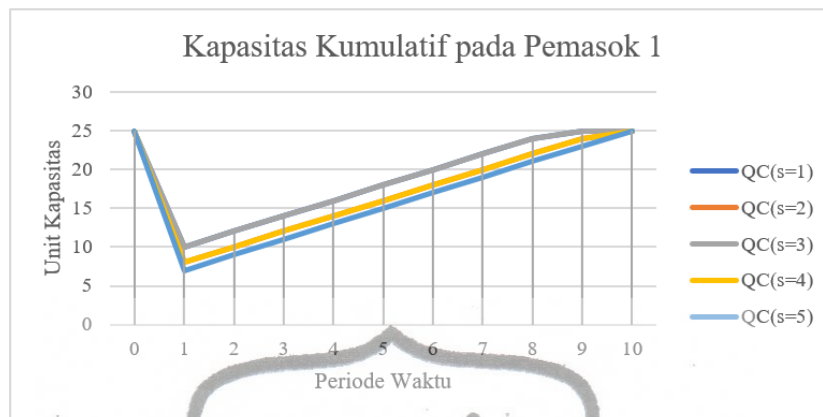
Gambar 4.16 Jumlah Kegagalan Kapasitas

Berikut adalah grafik jumlah kapasitas yang bertahan pada setiap skenario disrupsi dari solusi optimal yang didapatkan.



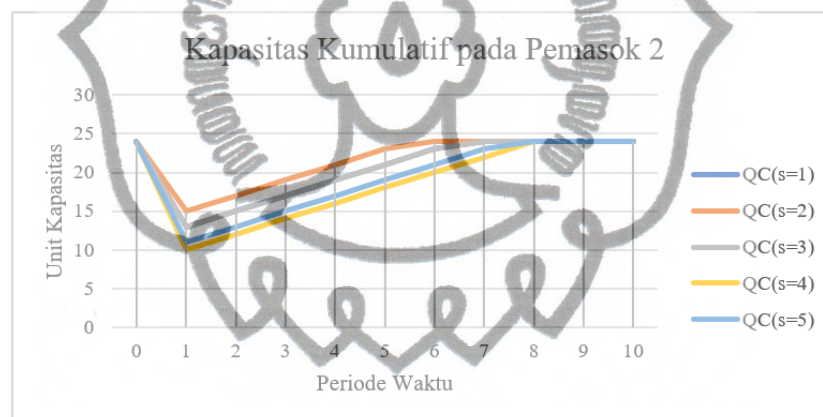
Gambar 4.17 Jumlah Kapasitas yang Bertahan

Berikut adalah grafik tingkat kapasitas produksi kumulatif untuk pemasok $i = 1$ dari solusi optimal yang didapatkan.



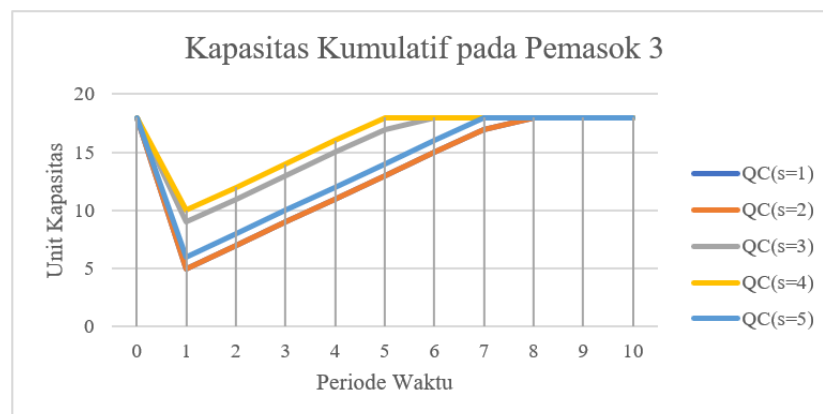
Gambar 4.18 Tingkat Kapasitas Produksi Kumulatif Pemasok 1

Berikut adalah grafik tingkat kapasitas produksi kumulatif untuk pemasok $i = 2$ dari solusi optimal yang didapatkan.



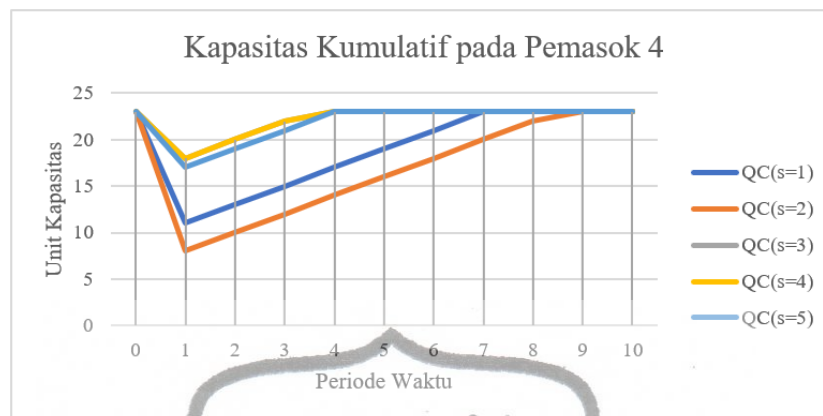
Gambar 4.19 Tingkat Kapasitas Produksi Kumulatif Pemasok 2

Berikut adalah grafik tingkat kapasitas produksi kumulatif untuk pemasok $i = 3$ dari solusi optimal yang didapatkan.



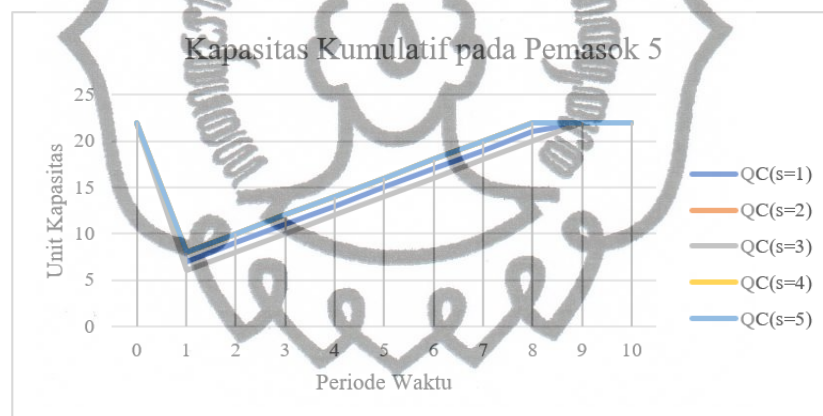
Gambar 4.20 Tingkat Kapasitas Produksi Kumulatif Pemasok 3

Berikut adalah grafik tingkat kapasitas produksi kumulatif untuk pemasok $i = 4$ dari solusi optimal yang didapatkan.



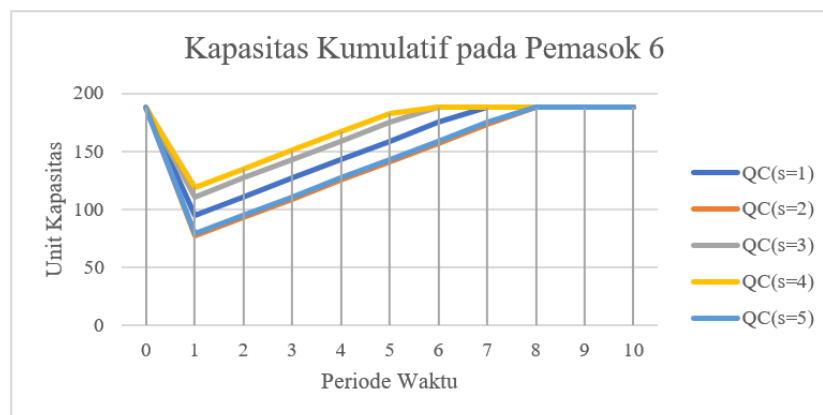
Gambar 4.21 Tingkat Kapasitas Produksi Kumulatif Pemasok 4

Berikut adalah grafik tingkat kapasitas produksi kumulatif untuk pemasok $i = 5$ dari solusi optimal yang didapatkan.



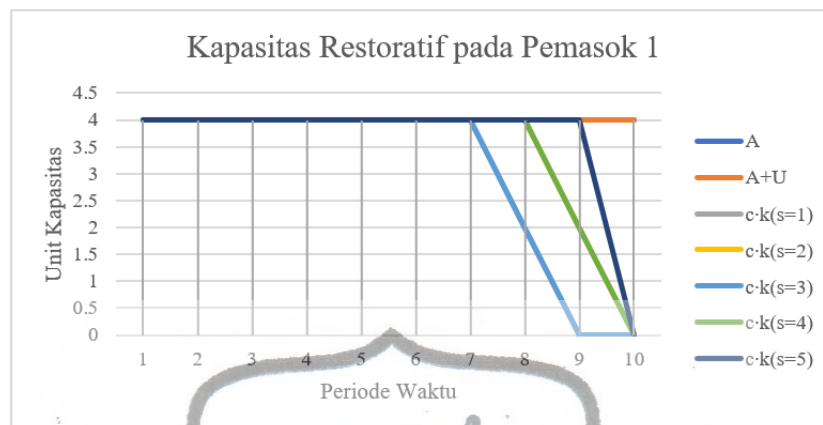
Gambar 4.22 Tingkat Kapasitas Produksi Kumulatif Pemasok 5

Berikut adalah grafik tingkat kapasitas produksi kumulatif untuk pemasok $i = 6$ dari solusi optimal yang didapatkan.



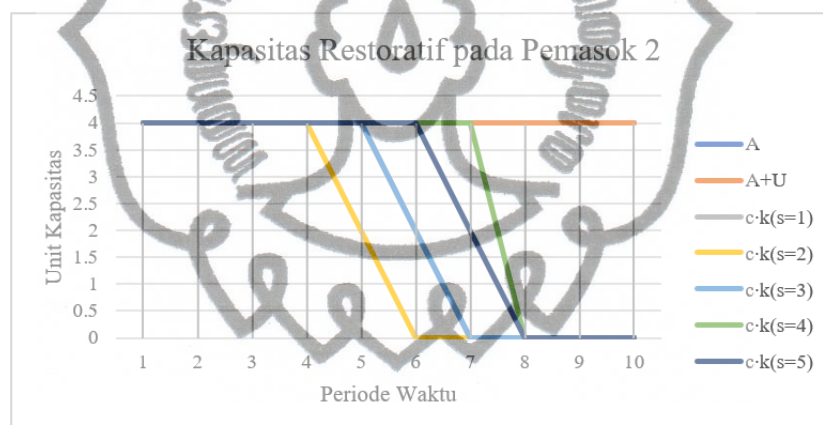
Gambar 4.23 Tingkat Kapasitas Produksi Kumulatif Pemasok 6

Berikut adalah grafik tingkat kapasitas restoratif untuk pemasok $i = 1$ dari solusi optimal yang didapatkan.



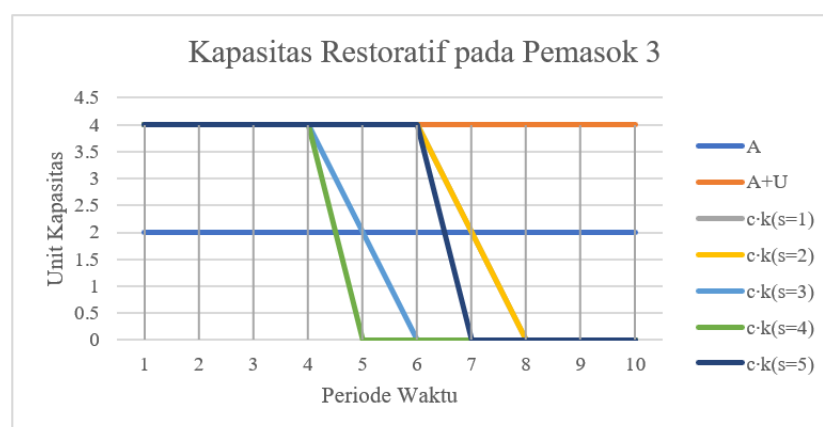
Gambar 4.24 Tingkat Kapasitas Restoratif Pemasok 1

Berikut adalah grafik tingkat kapasitas restoratif untuk pemasok $i = 2$ dari solusi optimal yang didapatkan.



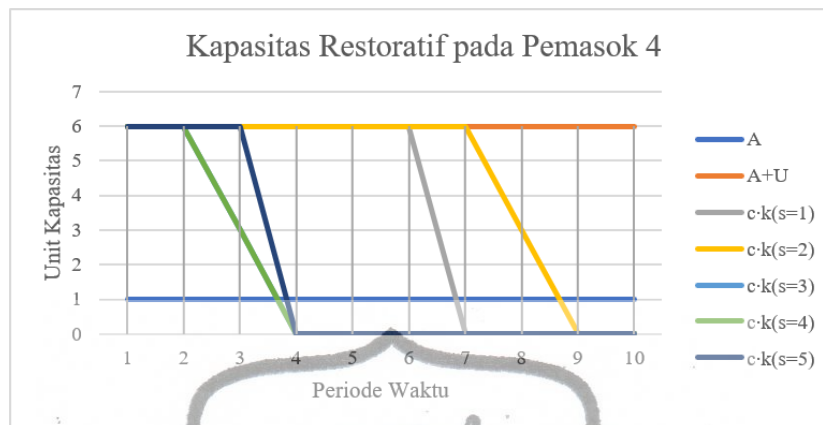
Gambar 4.25 Tingkat Kapasitas Restoratif Pemasok 2

Berikut adalah grafik tingkat kapasitas restoratif untuk pemasok $i = 3$ dari solusi optimal yang didapatkan.



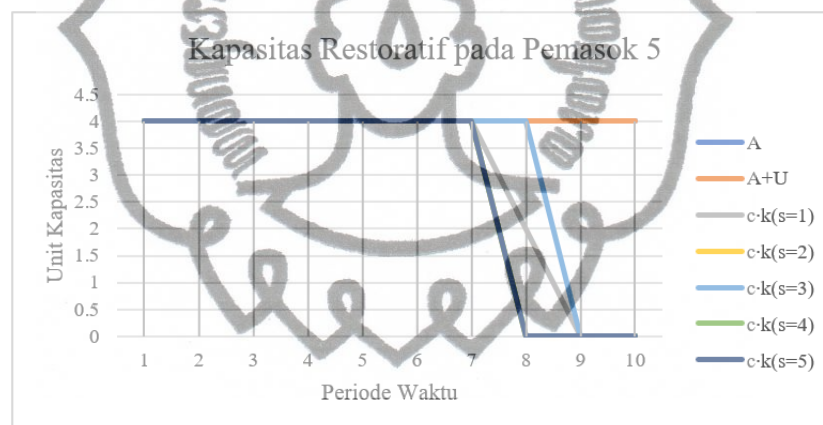
Gambar 4.26 Tingkat Kapasitas Restoratif Pemasok 3

Berikut adalah grafik tingkat kapasitas restoratif untuk pemasok $i = 4$ dari solusi optimal yang didapatkan.



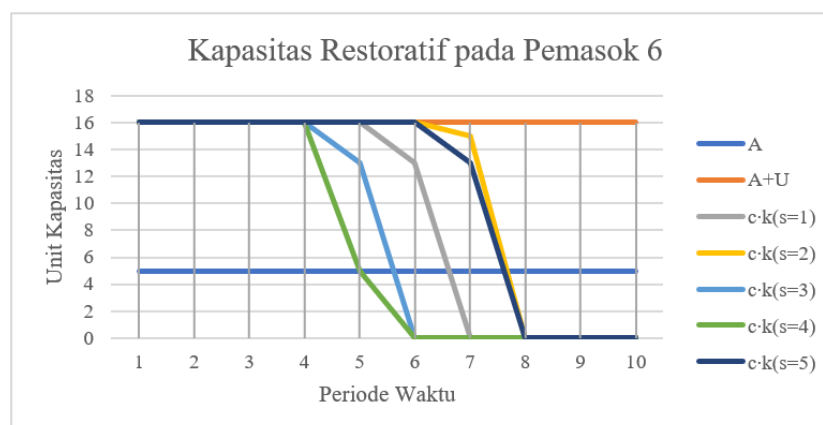
Gambar 4.27 Tingkat Kapasitas Restoratif Pemasok 4

Berikut adalah grafik tingkat kapasitas restoratif untuk pemasok $i = 5$ dari solusi optimal yang didapatkan.



Gambar 4.28 Tingkat Kapasitas Restoratif Pemasok 5

Berikut adalah grafik tingkat kapasitas restoratif untuk pemasok $i = 6$ dari solusi optimal yang didapatkan.



Gambar 4.29 Tingkat Kapasitas Restoratif Pemasok 6