

BAB III

PENYELESAIAN MASALAH PENUGASAN KARYAWAN DENGAN MENGUNAKAN ALGORITMA *FUZZY* EVOLUSI

Bab ini membahas tentang masalah penugasan karyawan pada suatu perusahaan jasa menggunakan algoritma *Fuzzy* Evolusi. Penugasan karyawan yang dimaksud adalah memasangkan pekerjaan pada karyawan yang ditentukan berdasarkan tingkat kemampuan karyawan selama periode waktu tertentu.

2.1 Deskripsi Masalah

Masalah penugasan karyawan yang dibahas pada penelitian ini dapat diilustrasikan sebagai berikut. Terdapat sejumlah karyawan yang akan ditugaskan ke sejumlah pekerjaan, dimana setiap karyawan memiliki tingkat kemampuan yang berbeda. Perbedaan tingkat kemampuan inilah akan menentukan jenis pekerjaan karyawan dan juga besaran gaji yang akan karyawan terima. Setiap karyawan diklasifikasikan dalam dua tipe karyawan. Pekerjaan jenis satu hanya bisa dikerjakan oleh karyawan tipe satu dan pekerjaan jenis dua dapat dikerjakan oleh karyawan tipe satu dan tipe dua. Setiap harinya terdapat daftar pekerjaan yang harus diselesaikan oleh karyawan. Penugasan karyawan akan dijadwalkan perminggu. Karyawan bekerja dari hari Senin sampai hari Minggu, dan karyawan dapat mengambil maksimal dua hari libur perminggu. Pemberian gaji dilakukan perminggu, dimana gaji karyawan tipe satu lebih besar dibandingkan gaji karyawan tipe dua. Tujuan dari penelitian ini adalah untuk menentukan penugasan karyawan yang optimal. Penugasan karyawan yang optimal adalah penugasan yang memasangkan karyawan pada pekerjaan yang ada dengan total gaji karyawan yang dikeluarkan adalah minimum.

Pada subbab selanjutnya akan dibahas model optimisasi masalah penugasan karyawan. Selanjutnya, model tersebut akan diselesaikan dengan menggunakan Algoritma *Fuzzy* Evolusi (FE). Penjelasan selengkapnya mengenai model penugasan karyawan dan cara kerja Algoritma *Fuzzy* Evolusi (FE) dijelaskan pada Subbab 3.2 dan Subbab 3.3.

2.2 Pemodelan masalah penugasan karyawan

Pada bagian ini akan diturunkan model optimisasi masalah penugasan karyawan. Adapun asumsi-asumsi yang ditentukan pada pemodelan ini adalah sebagai berikut:

1. Banyak karyawan, tipe pekerjaan, dan hari cukup untuk menugaskan semua karyawan yang ada
 2. Banyaknya tipe pekerjaan sama dengan banyaknya tipe karyawan
 3. Penelitian ini menentukan penugasan karyawan dalam satu minggu
 4. Semua karyawan adalah karyawan tetap dan dibedakan menjadi dua tipe, dengan tipe satu adalah karyawan yang memiliki tingkat kemampuan yang tinggi dan tipe dua adalah karyawan yang kemampuannya kurang dari tipe satu
 5. Perhitungan gaji karyawan bergantung pada tipe karyawan
 6. Setiap pekerjaan membutuhkan tepat satu karyawan dan karyawan tipe satu dapat menggantikan karyawan tipe dua tapi tidak berlaku sebaliknya
 7. Banyaknya hari libur karyawan dalam satu minggu maksimal adalah dua hari
- Tahapan pertama pemodelan adalah mendefinisikan himpunan, parameter dan variabel keputusan yang digunakan dalam model ini. Berikut adalah himpunan - himpunan yang digunakan dalam model. Misal:

J : himpunan hari

K : himpunan karyawan

L_j : himpunan pekerjaan pada hari ke j

Q : himpunan jenis pekerjaan

Adapun parameter-parameter model didefinisikan sebagai berikut, misal:

c_k : gaji karyawan ke k perminggu

$$z_{ljq} = \begin{cases} 1, & \text{jika pekerjaan ke } l \text{ pada hari ke } j \text{ termasuk dalam jenis pekerjaan ke } q \\ 0, & \text{yang lain} \end{cases}$$

Variabel keputusan dari model optimisasi didefinisikan untuk menentukan karyawan mana yang ditugaskan pada pekerjaan setiap harinya. Oleh karena itu variabel keputusan model optimisasi didefinisikan sebagai berikut. Misal:

$$x_{klj} = \begin{cases} 1, & \text{jika karyawan ke } k \text{ ditugaskan pada pekerjaan ke } l \text{ pada hari ke } j \\ 0, & \text{yang lain} \end{cases}$$

Kendala-kendala dari model optimisasi penugasan karyawan adalah hal-hal yang harus dipenuhi pada saat penyusunan tugas karyawan. Kendala-kendala tersebut adalah sebagai berikut:

1. Setiap pekerjaan harus dipasangkan tepat ke satu karyawan setiap harinya.

Kendala ini diekspresikan sebagai berikut:

$$\sum_{k \in K} x_{klj} = 1, \forall l_j \in L_j, \forall j \in J$$

2. Setiap karyawan mendapatkan maksimal 2 hari libur setiap minggunya. Ini berarti, dalam setiap minggu masing-masing karyawan bekerja minimal lima hari. Kendala ini diekspresikan sebagai berikut.

$$\sum_{j \in J} \sum_{l_j \in L_j} x_{klj} \geq 5, \forall k \in K$$

3. Setiap pekerjaan harus dipasangkan pada karyawan yang sesuai setiap harinya.

Kendala ini diekspresikan sebagai berikut.

$$\sum_{q \in Q} \sum_{k \in K} x_{klj} z_{ljq} = 1, \forall l_j \in L_j$$

Fungsi tujuan dari model optimisasi penugasan karyawan bertujuan untuk meminimumkan total gaji karyawan perminggu. Fungsi ini dapat diekspresikan sebagai berikut:

Meminimumkan:

$$z = \sum_{j \in J} \sum_{l \in L} \sum_{k \in K} C_k x_{klj} \quad (3.1)$$

berdasarkan

$$\sum_{k \in K} x_{klj} = 1, \forall l_j \in L_j, \forall j \in J \quad (3.2)$$

$$\sum_{j \in J} \sum_{l_j \in L_j} x_{klj} \geq 5, \forall k \in K \quad (3.3)$$

$$\sum_{q \in Q} \sum_{k \in K} x_{klj} z_{ljq} = 1, \forall l_j \in L_j \quad (3.4)$$

$$x_{klj} \in \{0,1\}, \forall l_j \in L_j, k \in K, j \in J \quad (3.5)$$

2.3 Implementasi Algoritma Fuzzy Evolusi pada Masalah Penugasan Karyawan

Masalah penugasan merupakan kasus khusus dari permasalahan transportasi. Permasalahan transportasi dapat digolongkan sebagai *NP-Hard Problem (Nondeterministic Polynomial Time)*. Pada *NP-Hard Problem* waktu yang dibutuhkan untuk mendapatkan solusi yang *feasible* dari permasalahan memakan waktu yang lama. Oleh karena itu, metode penyelesaian konvensional tidak efisien digunakan untuk menyelesaikan permasalahan sehingga diperlukan metode penyelesaian alternatif untuk menyelesaikan *NP-Hard Problem*.

Pada penelitian ini, untuk menyelesaikan model optimisasi masalah penugasan karyawan pada Subbab 3.2 peneliti menggunakan Algoritma FE (*Fuzzy Evolusi*). Algoritma FE adalah sebuah algoritma yang menggabungkan algoritma genetika dengan logika *fuzzy*. Algoritma FE biasa digunakan untuk menyelesaikan masalah optimisasi khususnya masalah penugasan. Pada algoritma FE, sebuah solusi pada persoalan penugasan direpresentasikan sebagai kromosom-kromosom. Sebuah kromosom terdiri dari sekumpulan gen yang merepresentasikan pekerjaan. Pada tahap awal dari algoritma FE, kromosom-kromosom ini dibangkitkan secara acak dalam sebuah populasi. Individu-individu dalam suatu populasi dapat direpresentasikan sebagai himpunan solusi penugasan. Kemudian himpunan solusi dievaluasi tingkat penyesuaian diri terhadap lingkungannya dengan sebuah nilai yang dinamakan sebagai nilai *fitness (fitness value)*. Nilai *fitness* ini biasanya berhubungan erat dengan fungsi tujuan dari permasalahan yang hendak diselesaikan. Semakin besar nilai *fitness* maka himpunan solusi tersebut akan memiliki peluang yang besar untuk menuju solusi optimum. Himpunan solusi yang mempunyai nilai *fitness* rendah akan dihapus dan diganti dengan himpunan solusi baru yang dibentuk berdasarkan informasi-informasi genetika dari himpunan sebelumnya yang mempunyai nilai *fitness* paling tinggi.

Himpunan solusi yang memiliki nilai *fitness* tinggi memiliki kesempatan untuk terseleksi menjadi induk dalam tahap reproduksi melalui perkawinan silang dengan solusi lain dalam populasi tersebut. Dalam tahap perkawinan silang, penentuan jumlah solusi yang akan melakukan perkawinan silang dilakukan dengan menentukan probabilitas perkawinan silang. Kemudian melakukan

pembangkitan bilangan acak, apabila bilangan acak yang dihasilkan lebih kecil dari ρc , maka solusi ini terpilih untuk melakukan perkawinan silang. Dalam menentukan nilai probabilitas perkawinan silang hal yang biasa dilakukan adalah mendefinisikan berdasarkan tetapan yang ada. Pada algoritma FE penentuan nilai dihasilkan dari logika *fuzzy*.

Pada tahap reproduksi perkawinan silang akan menghasilkan anak sebagai solusi baru, yang membawa beberapa sifat dari induknya. Sedangkan solusi dalam populasi yang tidak terseleksi dalam reproduksi akan mati dengan sendirinya dan tidak dapat digunakan kembali. Dengan cara seperti ini, beberapa generasi dengan karakteristik yang bagus akan bermunculan dalam populasi tersebut. Dengan banyak melakukan perkawinan silang, maka akan semakin banyak kemungkinan solusi terbaik yang dapat diperoleh.

Dari semua proses seleksi alam genetika, diharapkan menghasilkan solusi baru dengan nilai *fitness* yang lebih tinggi dari generasi sebelumnya sebagai generasi baru. Setelah didapat beberapa generasi maka akan didapatkan solusi yang optimal.

2.3.1 Representasi Kromosom

Satu kromosom merepresentasikan satu solusi, dalam hal ini, satu kromosom merupakan suatu penugasan karyawan dalam satu minggu. Satu kromosom memuat sejumlah gen, dimana satu gen mewakili satu slot pekerjaan. Jadi panjang satu kromosom sama dengan banyaknya slot pekerjaan selama satu minggu. Pada penelitian masalah optimisasi penugasan karyawan, peneliti merepresentasikan kromosom dengan menggunakan *value encoding*. Representasi kromosom dengan *value encoding* menggunakan kumpulan dari suatu nilai yang berupa macam-macam nilai sesuai dengan permasalahan yang dihadapi. Pada permasalahan ini nilai satu gen ke i mewakili karyawan yang ditugaskan pada pekerjaan ke i . Sebelumnya peneliti sudah membuat daftar pekerjaan yang harus dilakukan oleh karyawan setiap minggunya.

Panjang kromosom (banyaknya gen) menyatakan banyaknya pekerjaan dalam satu minggu. Terdapat dua jenis kromosom, jenis kromosom pertama untuk pekerjaan jenis satu dan kromosom jenis kedua untuk pekerjaan jenis dua.

Panjang dari kromosom jenis satu sama dengan jumlah pekerjaan ke satu dan panjang dari kromosom jenis dua sama dengan jumlah pekerjaan ke dua. Nilai setiap gen menyatakan identitas karyawan. Setiap pekerjaan dipasangkan pada karyawan yang sesuai. Contoh representasi kromosom untuk masalah penugasan karyawan pada pekerjaan dapat diilustrasikan pada Gambar 3.1 dan 3.2


Kromosom	4	2	4	3	1	4	3	2	4	3	2	3	2	3	3	1
1																

Gambar 3.1 Contoh Representasi Kromosom Jenis Pekerjaan Satu

Kromosom	2	6	5	2	4	2	4	7	1	6	1	5	5	6	7	1
1																
Kromosom	5	2	5	1	2	4	7	5	6	1	4	2	7	6	5	1
1																
Kromosom	7															
1																

Gambar 3.2 Contoh Representasi Kromosom Jenis Pekerjaan Dua

Keterangan:

	: hari Senin		: hari Kamis
	: hari Selasa		: hari Jumat
	: hari Rabu		: hari Sabtu
			: hari Minggu

2.3.2 Pembangkitan Populasi Awal

Proses pembangkitan populasi awal dilakukan secara acak. Populasi tersebut berisi beberapa kromosom yang telah didefinisikan sebelumnya. Banyaknya populasi awal yang dibangkitkan ditentukan oleh parameter *population_rate*.

2.3.3 Menghitung Nilai *Fitness*

Terdapat banyak cara untuk membangun nilai *fitness*. Untuk menghasilkan fungsi *fitness* yang baik, perlu dipahami batasan-batasan yang diberikan. Fungsi *fitness* ditentukan berdasarkan fungsi tujuan dari masalah optimisasi penugasan

karyawan. Tujuan dari penyelesaian masalah ini adalah untuk meminimumkan total gaji karyawan, fungsi *fitness* yang digunakan adalah

$$F = \frac{1}{z}$$

dimana F adalah fungsi *fitness* dan z adalah fungsi tujuan dari penugasan karyawan. Nilai z semakin kecil akan membuat fungsi *fitness* semakin besar dan sebaliknya.

2.3.4 Seleksi Populasi

Seleksi merupakan proses untuk memilih individu yang akan menjadi induk pada proses reproduksi. Pemilihan dilakukan dengan metode *roulette wheel*. Metode *roulette wheel* merupakan metode seleksi dengan cara memilih calon induk berdasarkan nilai *fitness* yang dimilikinya. Individu yang mempunyai *fitness value* terbaik mempunyai peluang lebih besar untuk menjadi induk dari individu berikutnya.

2.3.5 Penentuan parameter menggunakan fuzzy

Parameter pada algoritma genetika adalah ukuran populasi, banyaknya generasi, probabilitas *crossover* (ρc), dan probabilitas mutasi (ρm). Tidak adanya aturan tetap mengenai ukuran populasi dan banyaknya generasi menyebabkan perhitungan pada algoritma genetika harus dilakukan dalam beberapa kali percobaan hingga didapatkan solusi yang diharapkan. Untuk mengatasi permasalahan tersebut, dibutuhkan suatu metode yang dapat menghilangkan kesamaran dalam penentuan ukuran populasi dan banyaknya generasi. Dalam penelitian ini metode yang digunakan yaitu logika *fuzzy* metode Mamdani karena logika *fuzzy* dapat menghasilkan ρc dan ρm yang pasti. Pada logika *fuzzy* metode Mamdani, terdapat empat tahapan yang harus dilakukan untuk mendapatkan hasil (*output*). Tahap awal yang perlu dilakukan adalah menentukan variabel *input* dan variabel *output* berdasarkan data yang diperoleh, selanjutnya mengubahnya menjadi himpunan *fuzzy*, tahap awal ini disebut proses fuzzifikasi. Tahap selanjutnya adalah aplikasi fungsi implikasi. Pada tahap ini disusun basis aturan-aturan berupa implikasi-implikasi *fuzzy* yang menyatakan relasi antara variabel *input* dan variabel *output*. Tahap ketiga adalah komposisi aturan. Apabila logika

fuzzy terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan kolerasi antar aturan. Tahap terakhir adalah proses defuzzifikasi. Berdasarkan yang telah dilakukan pada tahap pertama, di mana data diubah menjadi himpunan *fuzzy* maka dalam prosesnya *output* pun berbentuk himpunan *fuzzy*, dalam proses ini himpunan *fuzzy* tersebut dikembalikan dalam bentuk suatu bilangan real yang tegas. Variabel *input* yang akan diproses menggunakan logika *fuzzy* metode mamdani adalah populasi dan generasi, sedangkan untuk *output* yang dihasilkan merupakan nilai dari ρc dan ρm yang optimal berdasarkan proses logika *fuzzy* metode Mamdani. Berikut merupakan penjelasan mengenai tahapan dari logika *fuzzy* metode Mamdani.

Sebelum memulai penjelasan mengenai tahapan dari logika *fuzzy* metode Mamdani, peneliti menentukan variabel input dan variabel output yang akan digunakan, dan juga menentukan aturan-aturan *fuzzy* untuk penentuan hasil.

- Variabel *input* yang digunakan adalah:
Populasi dan generasi
- Variabel *output* yang digunakan adalah:
Nilai probabilitas *crossover* (ρc) dan nilai probabilitas mutasi (ρm)

Aturan *fuzzy* yang digunakan dalam model Xu yang diutarakan oleh Eiben (dalam Muzid. S, 2008, hlm.C-34) didasarkan dari *input* jumlah populasi serta banyaknya generasi. Dari dua *input* tersebut akan menghasilkan nilai untuk ρc dan ρm . Aturan yang digunakan oleh Xu adalah

Tabel 3.1Aturan-aturan *fuzzy*

Aturan	Populasi	Generasi	ρc	ρm
1	Kecil	Pendek	Sedang	Besar
2	Sedang	Pendek	Kecil	Sedang
3	Besar	Pendek	Kecil	Kecil
4	Kecil	Sedang	Besar	Sedang
5	Sedang	Sedang	Besar	Kecil
6	Besar	Sedang	Sedang	Sangat Kecil
7	Kecil	Panjang	Sangat	Kecil

			Besar	
8	Sedang	Panjang	Besar	Sangat Kecil
Aturan	Populasi	Generasi	ρc	ρm
9	Besar	Panjang	Besar	Sangat Kecil

Dari tabel di atas, aturan-aturan *fuzzy* dapat ditulis sebagai berikut

1. IF (Populasi is Kecil) AND (Generasi is Pendek) THEN (ProbCrossover is Sedang) AND (ProbMutasi is Besar)
2. IF (Populasi is sedang) AND (Generasi is Pendek) THEN (ProbCrossover is Kecil) AND (ProbMutasi is Sedang)
3. IF (Populasi is Besar) AND (Generasi is Pendek) THEN (ProbCrossover is Kecil) AND (ProbMutasi is Kecil)
4. IF (Populasi is Kecil) AND (Generasi is Sedang) THEN (ProbCrossover is Besar) AND (ProbMutasi is Sedang)
5. IF (Populasi is Sedang) AND (Generasi is Sedang) THEN (ProbCrossover is Besar) AND (ProbMutasi is Kecil)
6. IF (Populasi is Besar) AND (Generasi is Sedang) THEN (ProbCrossover is Sedang) AND (ProbMutasi is Sangat Kecil)
7. IF (Populasi is Kecil) AND (Generasi is Panjang) THEN (ProbCrossover is Sangat Besar) AND (ProbMutasi is Kecil)
8. IF (Populasi is Sedang) AND (Generasi is Panjang) THEN (ProbCrossover is Sangat Besar) AND (ProbMutasi is Sangat Kecil).
9. IF (Populasi is Besar) AND (Generasi is Panjang) THEN (ProbCrossover is Besar) AND (ProbMutasi is Sangat Kecil).

Aturan- aturan yang dikembangkan oleh Xu akan digunakan dalam logika *fuzzy* metode Mamdani, tetapi perlu diperhatikan agar logika *fuzzy* dapat menghasilkan *output* maka diperlukan semesta pembicaraan dan domain yang memberikan nilai batas untuk setiap himpunan yang ada pada setiap variabel.

2.3.5.1 Fuzzifikasi

Tahap pertama yaitu menentukan semua variabel yang terkait dalam proses pengoptimalan. Variabel tersebut berupa dua variabel *input* yaitu variabel populasi dan generasi dan dua variabel *output* yaitu variabel pc dan pm . Variabel tersebut kemudian diubah menjadi himpunan *fuzzy* dengan membagi masing-masing variabel menjadi satu atau lebih himpunan *fuzzy*. Untuk variabel ukuran populasi dibagi menjadi tiga himpunan *fuzzy* yaitu kecil, sedang dan besar, untuk variabel banyaknya generasi dibagi menjadi tiga himpunan *fuzzy* yaitu pendek, sedang dan panjang. Kemudian akan ditentukan suatu fungsi fuzzifikasi yang sesuai. Fungsi yang dimaksud disini adalah suatu nilai yang menyatakan tingkat keanggotaan suatu variabel dalam suatu himpunan *fuzzy* dan disebut fungsi keanggotaan (μ).

a. Variabel Populasi

Pada semesta pembicaraan dan domain untuk variabel populasi, nilai yang digunakan sebagai berikut:

Semesta pembicaraan: $\{x: x \in \mathbb{R} \mid 0 \leq x \leq 100\}$

i. Domain kecil : $\{x: x \in \mathbb{R} \mid 10 \leq x \leq 35\}$

Fungsi Keanggotaan (μ)

$$\mu_{Popkecil}(x) = \begin{cases} 1 & ; x \leq 10 \\ 1 - 2 \left(\frac{x - 10}{35 - 10} \right)^2 & ; 10 < x \leq 22,5 \\ 2 \left(\frac{35 - x}{35 - 10} \right)^2 & ; 22,5 < x < 35 \\ 0 & ; x \geq 35 \end{cases}$$

ii. Domain sedang: $\{x: x \in \mathbb{R} \mid 15 \leq x \leq 50\}$

Fungsi Keanggotaan (μ)

$$\mu_{Popsedang}(x) = \exp\left(\frac{-(x - 50)^2}{2(15)^2}\right)$$

iii. Domain besar: $\{x: x \in \mathbb{R} \mid 65 \leq x \leq 90\}$

Fungsi Keanggotaan (μ)

$$\mu_{Popbesar}(x) = \begin{cases} 0 & ; x \leq 65 \\ 2 \left(\frac{x-65}{90-65} \right)^2 & ; 65 < x \leq 77,5 \\ 1 - 2 \left(\frac{90-x}{90-65} \right)^2 & ; 77,5 < x < 90 \\ 1 & ; x \geq 90 \end{cases}$$

b. Variabel Generasi

Pada semesta pembicaraan dan domain untuk variabel populasi, nilai yang digunakan sebagai berikut:

Semesta pembicaraan: $\{x: x \in \mathbb{R} \mid 0 \leq x \leq 100\}$

i. Domain pendek: $\{x: x \in \mathbb{R} \mid 10 \leq x \leq 35\}$

Fungsi Keanggotaan (μ)

$$\mu_{Genpendek}(x) = \begin{cases} 1 & ; x \leq 10 \\ 1 - 2 \left(\frac{x-10}{35-10} \right)^2 & ; 10 < x \leq 22,5 \\ 2 \left(\frac{35-x}{35-10} \right)^2 & ; 22,5 < x < 35 \\ 0 & ; x \geq 35 \end{cases}$$

ii. Domain sedang: $\{x: x \in \mathbb{R} \mid 15 \leq x \leq 50\}$

Fungsi Keanggotaan (μ)

$$\mu_{Gensedang}(x) = \exp\left(\frac{-(x-50)^2}{2(15)^2}\right)$$

iii. Domain panjang: $\{x: x \in \mathbb{R} \mid 65 \leq x \leq 90\}$

Fungsi Keanggotaan (μ)

$$\mu_{Genpanjang}(x) = \begin{cases} 0 & ; x \leq 65 \\ 2 \left(\frac{x-65}{90-65} \right)^2 & ; 65 < x \leq 77,5 \\ 1 - 2 \left(\frac{90-x}{90-65} \right)^2 & ; 77,5 < x < 90 \\ 1 & ; x \geq 90 \end{cases}$$

c. Variabel Probabilitas *Crossover*

Pada umumnya nilai probabilitas *crossover* adalah antara 0,6 sampai 0,9 sehingga pada semesta pembicaraan dan domain untuk hasil *output* yaitu nilai probabilitas *crossover*, aturan nilai yang digunakan adalah sebagai berikut:

Semesta pembicaraan: $\{x: x \in \mathbb{R} \mid 0,6 \leq x \leq 0,9\}$

i. Domain kecil: $\{x: x \in \mathbb{R} \mid 0,625 \leq x \leq 0,7\}$

Fungsi Keanggotaan (μ)

$$\mu_{pkecil}(x) = \begin{cases} 1 & ; \quad x \leq 0,625 \\ 1 - 2 \left(\frac{x - 0,625}{0,7 - 0,625} \right)^2 & ; \quad 0,625 < x \leq 0,6625 \\ 2 \left(\frac{0,7 - x}{0,7 - 0,625} \right)^2 & ; \quad 0,6625 < x < 0,7 \\ 0 & ; \quad x \geq 0,7 \end{cases}$$

ii. Domain sedang: $\{x: x \in \mathbb{R} \mid 0,63 \leq x \leq 0,78\}$

Fungsi Keanggotaan (μ)

$$\mu_{psedang}(x) = \begin{cases} 0 & ; \quad x \leq 0,63 \text{ atau } x \geq 0,78 \\ \frac{x - 0,63}{0,7 - 0,63} & ; \quad 0,63 < x < 0,7 \\ 1 & ; \quad 0,7 \leq x \leq 0,72 \\ \frac{0,78 - x}{0,78 - 0,72} & ; \quad 0,72 < x < 0,78 \end{cases}$$

iii. Domain Besar: $\{x: x \in \mathbb{R} \mid 0,72 \leq x \leq 0,87\}$

Fungsi Keanggotaan (μ)

$$\mu_{pbesar}(x) = \begin{cases} 0 & ; \quad x \leq 0,72 \text{ atau } x \geq 0,87 \\ \frac{x - 0,72}{0,78 - 0,72} & ; \quad 0,72 < x < 0,78 \\ 1 & ; \quad 0,78 \leq x \leq 0,8 \\ \frac{0,87 - x}{0,87 - 0,8} & ; \quad 0,8 < x < 0,87 \end{cases}$$

iv. Domain Sangat Besar: $\{x: x \in \mathbb{R} \mid 0,8 \leq x \leq 0,875\}$

Fungsi Keanggotaan (μ)

$$\mu_{psangatbesar}(x) = \begin{cases} 0 & ; \quad x \leq 0,8 \\ 2 \left(\frac{x - 0,8}{0,875 - 0,8} \right)^2 & ; \quad 0,8 < x \leq 0,8375 \\ 1 - 2 \left(\frac{0,875 - x}{0,875 - 0,8} \right)^2 & ; \quad 0,8375 < x < 0,875 \\ 1 & ; \quad x \geq 0,875 \end{cases}$$

d. Variabel Probabilitas Mutasi

Probabilitas mutasi pada umumnya sangat kecil yaitu sekitar 1 dibagi jumlah gen yang digunakan. Artinya probabilitas mutasi hanya terjadi pada kisaran satu gen saja pada tiap individu atau dengan kata lain probabilitas mutasi mendekati nol. Sehingga pada semesta pembicaraan dan domain untuk nilai probabilitas mutasi, aturan nilai yang digunakan adalah sebagai berikut:

Semesta pembicaraan: $\{x: x \in \mathbb{R} \mid 0 \leq x \leq 0,25\}$

i. Domain Sangat Kecil: $\{x: x \in \mathbb{R} \mid 0,025 \leq x \leq 0,1\}$

Fungsi Keanggotaan (μ)

$$\mu_{\rho\text{msangatkecil}}(x) = \begin{cases} 1 & ; \quad x \leq 0,025 \\ 1 - 2 \left(\frac{x - 0,025}{0,1 - 0,025} \right)^2 & ; \quad 0,025 < x \leq 0,0625 \\ 2 \left(\frac{0,1 - x}{0,1 - 0,025} \right)^2 & ; \quad 0,0625 < x < 0,1 \\ 0 & ; \quad x \geq 0,1 \end{cases}$$

ii. Domain kecil: $\{x: x \in \mathbb{R} \mid 0,047 \leq x \leq 0,14\}$

Fungsi Keanggotaan (μ)

$$\mu_{\rho\text{mkecil}}(x) = \begin{cases} 0 & ; \quad x \leq 0,047 \text{ atau } x \geq 0,14 \\ \frac{x - 0,047}{0,083 - 0,047} & ; \quad 0,047 < x < 0,083 \\ 1 & ; \quad 0,083 \leq x \leq 0,1 \\ \frac{0,14 - x}{0,14 - 0,1} & ; \quad 0,1 < x < 0,14 \end{cases}$$

iii. Domain Sedang: $\{x: x \in \mathbb{R} \mid 0,1 \leq x \leq 0,2\}$

Fungsi Keanggotaan (μ)

$$\mu_{\rho\text{mse dang}}(x) = \begin{cases} 0 & ; \quad x \leq 0,1 \text{ atau } x \geq 0,2 \\ \frac{x - 0,1}{0,14 - 0,1} & ; \quad 0,1 < x < 0,14 \\ 1 & ; \quad 0,14 \leq x \leq 0,167 \\ \frac{0,2 - x}{0,2 - 0,167} & ; \quad 0,167 < x < 0,2 \end{cases}$$

iv. Domain Besar: $\{x: x \in \mathbb{R} \mid 0,15 \leq x \leq 0,225\}$

Fungsi Keanggotaan (μ)

$$\mu_{pmbesar}(x) = \begin{cases} 0 & ; x \leq 0,15 \\ 2 \left(\frac{x - 0,15}{0,225 - 0,15} \right)^2 & ; 0,15 < x \leq 0,1875 \\ 1 - 2 \left(\frac{0,225 - x}{0,225 - 0,15} \right)^2 & ; 0,1875 < x < 0,225 \\ 1 & ; x \geq 0,225 \end{cases}$$

Apabila dipilih populasi sebesar 20 diperoleh,

$$\mu_{Popkecil}(20) = 1 - 2 \left(\frac{20-10}{35-10} \right)^2 = 1 - 2 \frac{4}{25} = 0,68$$

$$\mu_{Popsedang}(20) = \exp \left(\frac{-(20-50)^2}{2(15)^2} \right) = \exp \left(\frac{-900}{450} \right) = e^{-2} = 0,135$$

Perhitungan di atas menyatakan bahwa populasi sebesar 20 tergolong himpunan kecil dengan derajat keanggotaan 0,68 atau tergolong himpunan sedang dengan nilai derajat keanggotaan 0,135.

Jika dipilih generasi sebesar 50 diperoleh,

$$\mu_{Gensedang}(50) = \exp \left(\frac{-(50-50)^2}{2(15)^2} \right) = \exp \left(\frac{0}{450} \right) = e^0 = 1$$

Perhitungan di atas menyatakan bahwa generasi sebesar 50 tergolong himpunan sedang dengan nilai derajat keanggotaan 1.

2.3.5.2 Aplikasi Fungsi Implikasi

Langkah selanjutnya adalah penyusunan basis aturan, yaitu aturan-aturan berupa implikasi-implikasi *fuzzy* yang menyatakan relasi antara variabel *input* yaitu variabel populasi dan generasi untuk memperoleh nilai tunggal dari variabel *output* yaitu variabel *pc* dan variabel *pm*. Penentuan variabel *pc* dan variabel *pm* dilakukan dengan memperhatikan hubungan irisan dari variabel-variabel populasi dan generasi, maka aturan yang digunakan pada fungsi adalah aturan *Min* dengan kata lain operator *fuzzy* yang digunakan adalah operator AND. Bentuk umumnya adalah sebagai berikut:

jika a adalah A dan b adalah B, maka c adalah C

dengan A, B, dan C adalah predikat-predikat *fuzzy* yang merupakan nilai linguistik dari masing-masing variabel. Banyaknya aturan ditentukan oleh banyaknya nilai linguistik untuk masing-masing variabel *input*. Misalkan pada permasalahan contoh $\mu - predikat_i$ adalah fungsi implikasi ke-i, di mana:

$$\mu - \text{predikat}_1 = \min(\mu_R(x), \mu_R(y), \mu_R(z))$$

$$\mu - \text{predikat}_2 = \min(\mu_R(x), \mu_R(y), \mu_L(z))$$

$$\mu - \text{predikat}_3 = \min(\mu_R(x), \mu_L(y), \mu_R(z))$$

$$\mu - \text{predikat}_4 = \min(\mu_R(x), \mu_L(y), \mu_L(z))$$

$$\mu - \text{predikat}_5 = \min(\mu_L(x), \mu_R(y), \mu_R(z))$$

$$\mu - \text{predikat}_6 = \min(\mu_L(x), \mu_R(y), \mu_L(z))$$

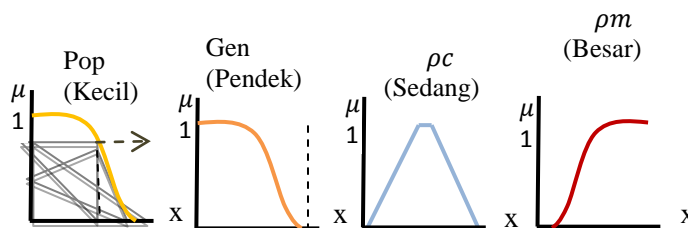
$$\mu - \text{predikat}_7 = \min(\mu_L(x), \mu_L(y), \mu_R(z))$$

$$\mu - \text{predikat}_8 = \min(\mu_L(x), \mu_L(y), \mu_L(z))$$

Selanjutnya, jika diketahui populasi 20 dan generasi 50, maka:

- a) [R1] IF(Populasi) kecil AND (Generasi) pendek THEN (ρc) sedang AND (ρm) besar

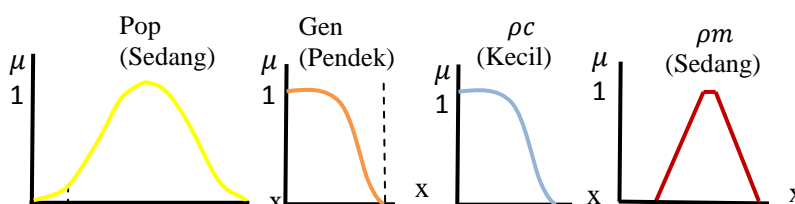
$$\begin{aligned} \alpha - \text{predikat}_1 &= \mu_{\mu_{\text{popkecil}}} \cdot \mu_{\text{genpendek}} \\ &= \min(\mu_{\text{popkecil}}(20), \mu_{\text{genpendek}}(50)) \\ &= \min(0,68 ; 0) \\ &= 0 \end{aligned}$$



Gambar 3.3 Ilustrasi Fungsi Implikasi :Min [R1]

- b) [R2] IF(Populasi) sedang AND (Generasi) pendek THEN (ρc) kecil AND (ρm) sedang

$$\begin{aligned} \alpha - \text{predikat}_2 &= \mu_{\text{popsedang}} \cdot \mu_{\text{genpendek}} \\ &= \min(\mu_{\text{popsedang}}(20), \mu_{\text{genpendek}}(50)) \\ &= \min(0,135 ; 0) \\ &= 0 \end{aligned}$$



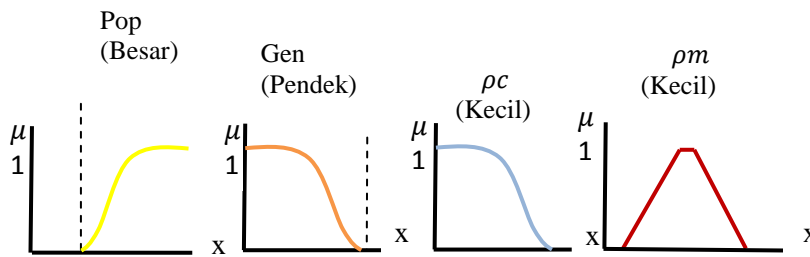
Tamara Widiyanti, 2016



Gambar 3.4 Ilustrasi Fungsi Implikasi :*Min* [R2]

- c) [R3] IF(Populasi) besar AND (Generasi) pendek THEN (ρc) kecil AND (ρm) kecil

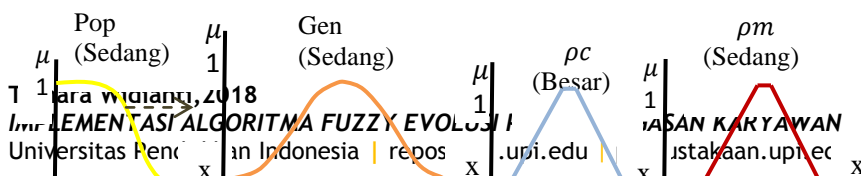
$$\begin{aligned} \alpha - \text{predikat}_3 &= \mu_{\text{popbesar}} , \mu_{\text{genpendek}} \\ &= \min(\mu_{\text{popbesar}}(20), \mu_{\text{genpendek}}(50)) \\ &= \min(0; 0) \\ &= 0 \end{aligned}$$

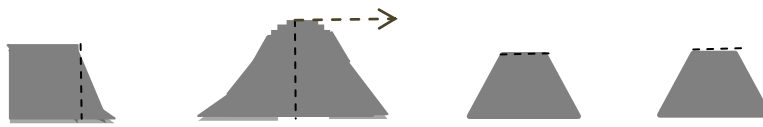


Gambar 3.5 Ilustrasi Fungsi Implikasi :*Min* [R3]

- d) [R4] IF(Populasi) kecil AND (Generasi) sedang THEN (ρc) besar AND (ρm) sedang

$$\begin{aligned} \alpha - \text{predikat}_4 &= \mu_{\text{popkecil}} , \mu_{\text{gensedang}} \\ &= \min(\mu_{\text{popkecil}}(20), \mu_{\text{gensedang}}(50)) \\ &= \min(0,68; 1) \\ &= 0,68 \end{aligned}$$

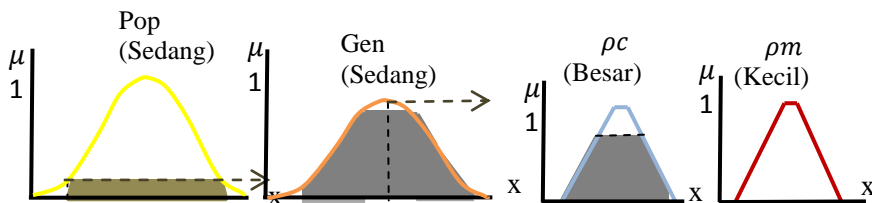




Gambar 3.6 Ilustrasi Fungsi Implikasi :*Min* [R4]

- e) [R5] IF(Populasi) sedang AND (Generasi) sedang THEN (ρc) besar AND (ρm) kecil

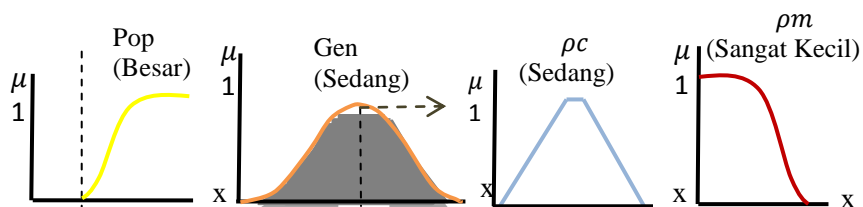
$$\begin{aligned}
 \alpha - \text{predikat}_5 &= \mu_{\text{popsedang}} , \mu_{\text{gensedang}} \\
 &= \min(\mu_{\text{popsedang}}(20), \mu_{\text{gensedang}}(50)) \\
 &= \min(0,135 ; 1) \\
 &= 0,135
 \end{aligned}$$



Gambar 3.7 Ilustrasi Fungsi Implikasi :*Min* [R5]

- f) [R6] IF(Populasi) besar AND (Generasi) sedang THEN (ρc) sedang AND (ρm) sangat kecil

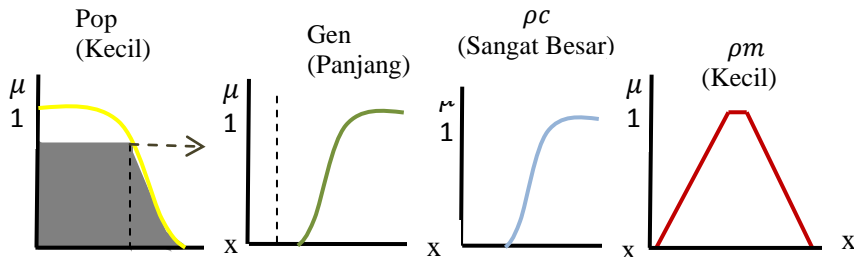
$$\begin{aligned}
 \alpha - \text{predikat}_6 &= \mu_{\text{popbesar}} , \mu_{\text{gensedang}} \\
 &= \min(\mu_{\text{popbesar}}(20), \mu_{\text{gensedang}}(50)) \\
 &= \min(0 ; 1) \\
 &= 0
 \end{aligned}$$



Gambar 3.8 Ilustrasi Fungsi Implikasi :*Min* [R6]

- g) [R7] IF(Populasi) kecil AND (Generasi) panjang THEN (ρc) sangat besar AND (ρm) kecil

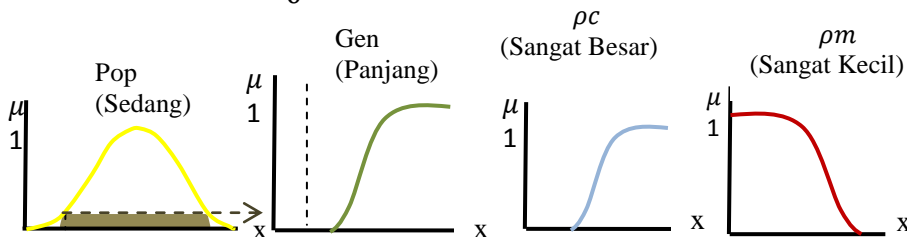
$$\begin{aligned}\alpha - \text{predikat}_7 &= \mu_{\text{popkecil}} , \mu_{\text{genpanjang}} \\ &= \min(\mu_{\text{popkecil}}(20), \mu_{\text{genpanjang}}(50)) \\ &= \min(0,68 ; 0) \\ &= 0\end{aligned}$$



Gambar 3.9 Ilustrasi Fungsi Implikasi : *Min* [R7]

- h) [R8] IF(Populasi) sedang AND (Generasi) panjang THEN (ρc) sangat besar AND (ρm) sangat kecil

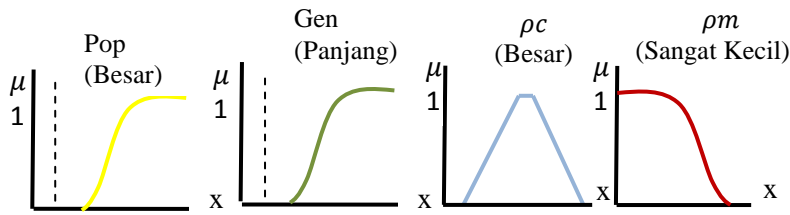
$$\begin{aligned}\alpha - \text{predikat}_8 &= \mu_{\text{popsedang}} , \mu_{\text{genpanjang}} \\ &= \min(\mu_{\text{popsedang}}(20), \mu_{\text{genpanjang}}(50)) \\ &= \min(0,135 ; 0) \\ &= 0\end{aligned}$$



Gambar 3.10 Ilustrasi Fungsi Implikasi : *Min* [R8]

- i) [R9] IF(Populasi) besar AND (Generasi) panjang THEN (ρc) besar AND (ρm) sangat kecil

$$\begin{aligned}\alpha - \text{predikat}_9 &= \mu_{\text{popbesar}} , \mu_{\text{genpanjang}} \\ &= \min(\mu_{\text{popbesar}}(20), \mu_{\text{genpanjang}}(50)) \\ &= \min(0 ; 0) \\ &= 0\end{aligned}$$



Gambar 3.11 Ilustrasi Fungsi Implikasi :Min [R9]

3.2.5.3 Komposisi Aturan

Pada penelitian ini metode komposisi aturan yang dipakai adalah metode *Max*. Pada metode *Max*, solusi himpunan *fuzzy* diperoleh dengan mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy* dan mengaplikasikannya ke *output* dengan menggunakan operator OR (*union*). Jika semua proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* dengan merefleksikan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan sebagai berikut :

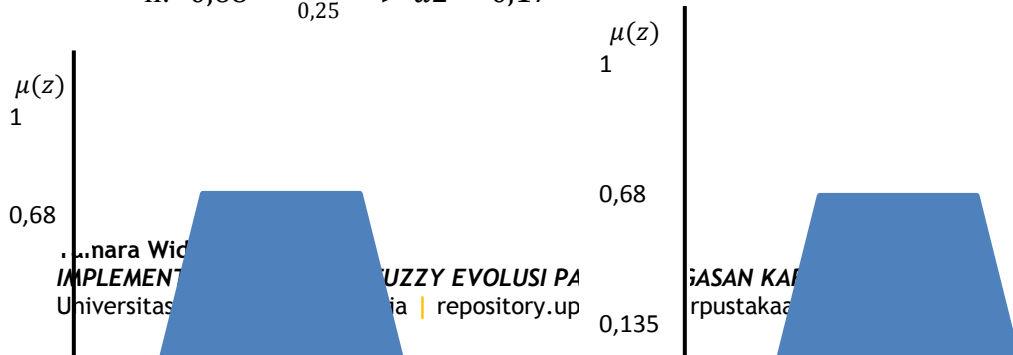
$$\mu_{sf}(x_i) = \max(\mu_{sf}(x_i), \mu_{kf}(x_i))$$

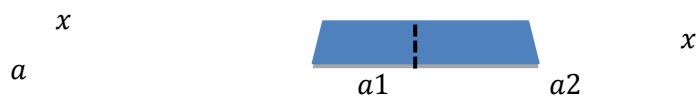
dengan $\mu_{sf}(x_i)$ adalah nilai keanggotaan solusi *fuzzy* sampai aturan ke-i, dan $\mu_{kf}(x_i)$ adalah nilai keanggotaan konsekuen *fuzzy* aturan ke-i. Misalkan $\mu(z)$ mewakili hasil dari semua α – predikat

Berdasarkan daerah hasil komposisi pada Gambar 3.11 akan dicari nilai-nilai a untuk ρc dan a_1 a_2 untuk ρm . Nilai –nilai tersebut diperoleh berdasarkan fungsi keanggotan ρc dan ρm yang telah ditetapkan sebelumnya, yaitu sebagai berikut :

- Untuk ρc

$$0,68 = \frac{a - 0,6}{0,3} \Rightarrow a = 0,804$$
- Untuk ρm
 - i. $0,135 = \frac{a_1 - 0}{0,25} \Rightarrow a_1 = 0,034$
 - ii. $0,68 = \frac{a_2 - 0}{0,25} \Rightarrow a_2 = 0,17$





Gambar 3.12 Daerah Hasil Komposisi ρc dan ρm

2.3.5.3 Defuzzifikasi

Setelah data melalui tahapan fuzzifikasi data bukan lagi dalam bentuk tegas, namun dalam bentuk himpunan *fuzzy*, maka dalam proses pencarian keputusan tersebut diperoleh data *output* yang berbentuk himpunan *fuzzy* pula, pada tahap ini data yang berbentuk himpunan *fuzzy* tersebut diubah kembali kedalam bentuk tegas. Data tersebut merupakan data optimal berdasarkan logika *fuzzy* metode Mamdani. Terdapat beberapa metode defuzzifikasi pada komposisi aturan metode Mamdani, antara lain adalah metode centroid, metode bisektor, metode *mean of maximum*, metode *largest of maximum*, dan metode *smallest of maximum*.

Pada penelitian ini, metode yang dilakukan dalam proses defuzzifikasi adalah Metode Centroid (*Composite Moment*). Pada metode ini, peneliti menganggap bahwa variabel yang ada merupakan bilangan diskrit. Misalkan z adalah nilai ρc dan ρm yang diperoleh pada proses komposisi aturan, maka z^* merupakan ρc dan ρm optimal secara umum dirumuskan :

$$z^* = \frac{\sum_{i=1}^n z_i \mu(z_i)}{\sum_{i=1}^n \mu(z_i)}$$

Untuk kasus populasi 20 dan generasi 50, maka diperoleh ρc dan ρm optimal (z^*) yang dicari adalah sebagai berikut:

- Untuk ρc

$$z^* = \frac{0,804 * 0,68}{0,68}$$

$$z^* = 0,804$$

- Untuk ρm

$$z^* = \frac{(0,135 * 0,034) + (0,68 * 0,17)}{0,135 + 0,68}$$

$$z^* = \frac{0,12}{0,815}$$

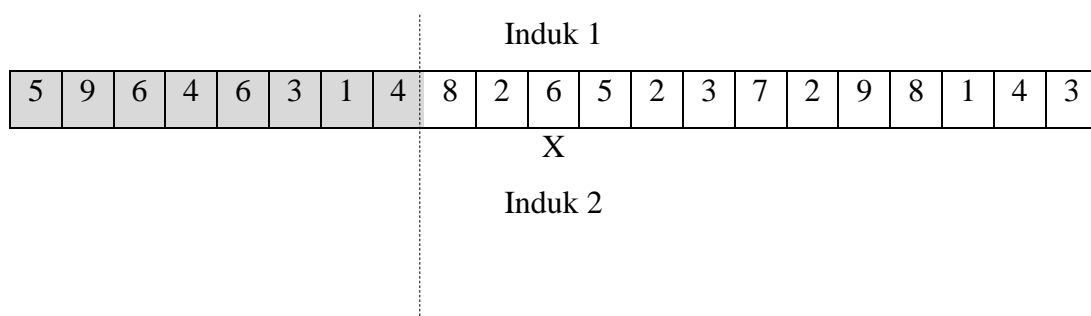
$$z^* = 0,147$$

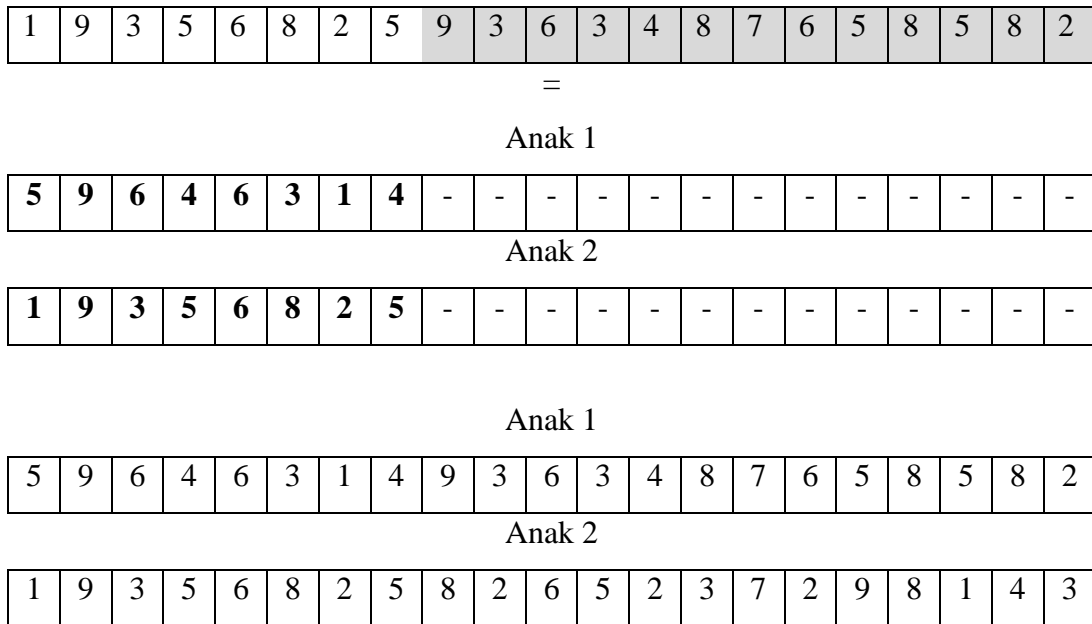
2.3.6 Crossover

Kromosom yang telah diperoleh dari proses seleksi selanjutnya akan direkombinasi (*crossover*). *Crossover* adalah operator yang membutuhkan dua kromosom atau biasa disebut kromosom induk untuk menghasilkan kromosom baru sebagai anak. Untuk masalah penugasan karyawan metode *crossover* yang digunakan adalah metode *single point crossover* untuk representasi integer. Proses ini ditentukan oleh parameter *crossover_rate* (ρc) yang telah didapatkan dari logika *fuzzy* mamdani dan akan dibangkitkan bilangan acak 0 sampai 1. Berikut adalah langkah-langkah dari proses *crossover*,

1. Tentukan dua kromosom yang akan dijadikan induk 1 dan induk 2 dari proses seleksi dengan membandingkan bilangan acak yang dibangkitkan dengan parameter *crossover_rate* (ρc). Jika nilai bilangan acak kurang dari ρc maka kromosom terpilih menjadi induk dan jika tidak maka sebaliknya.
2. Tentukan titik *crossover* caranya dengan pilih satu nilai integer secara acak dari 1 sampai n banyaknya pekerjaan. Misal terpilih titik potong *crossover* pada posisi pekerjaan ke 8 dari kromosom pada Gambar 3.13.
3. Selanjutnya, buat kromosom kosong sebagai anak 1 dan anak 2 kemudian menempatkan gen-gen sebelum titik potong tersebut pada induk 1 kedalam ruang gen anak 1, dan selanjutnya mengisi ruang kosong yang lain dengan gen dari induk 2 setelah titik potong. Begitu pula dilakukan hal yang sama untuk kromosom anak 2.
4. Setelah seluruh gen terisi, maka anak 1 dan anak 2 merupakan kromosom baru hasil *crossover*.

Proses *crossover* dari dua buah induk kromosom diilustrasikan pada Gambar 3.3.





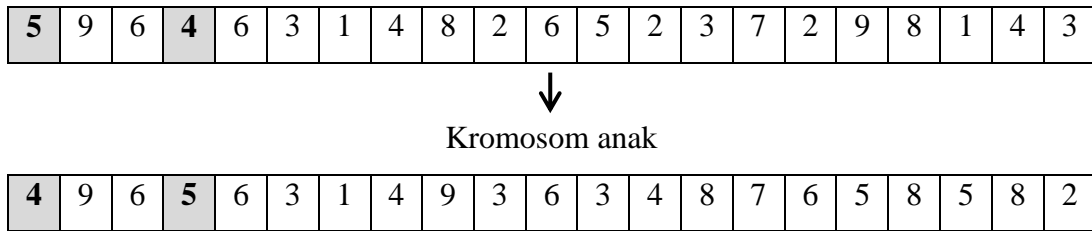
Gambar 3.13 Single Point crossover

2.3.7 Mutasi

Pada penelitian ini jenis mutasi yang digunakan adalah *swapping mutation*. Jumlah kromosom induk yang mengalami mutasi adalah seluruh kromosom baru hasil *crossover*. Gen yang akan dimutasi ditentukan oleh parameter yang dinamakan *mutation_rate* (ρm). Proses mutasi dilakukan dengan cara mengganti posisi gen yang terpilih dengan posisi gen lainnya yang dipilih secara acak.

Dari logika *fuzzy* misal diperoleh ρm sebesar 0,1 dan gen menyatakan karyawan. Untuk setiap karyawan yang ada akan di bangkitkan nilai integer acak dari 0 sampai 1. Jika pada urutan pekerjaan pertama terpilih nilai bilangan acak 0,05 berarti nilai tersebut kurang dari 0,1 maka urutan pekerjaan pertama akan ditukarkan dengan pekerjaan urutan yang lain yang dipilih secara acak misal urutan ke-4 maka urutan pekerjaan pertama ditukar dengan urutan pekerjaan ke-4. Untuk urutan pekerjaan ke-2 sampai ke-9 terpilih bilangan acak lebih dari 0,1 maka tidak terjadi proses mutasi. Tujuan dari mutasi ini untuk mendapat nilai *fitness* yang lebih baik sehingga mendapat penugasan terbaik dalam penelitian ini. Hasil dari mutasi diilustrasikan pada Gambar 3.14.

Kromosom induk



Gambar 3.14 *Swapping Mutation*

Setelah menyelesaikan proses mutasi maka satu generasi atau iterasi telah terselesaikan sehingga diperoleh kromosom baru hasil seleksi, *crossover* dan mutasi. Dengan adanya kromosom baru maka ukuran populasi menjadi dua kali lipat. Setiap kromosom tersebut akan dihitung kembali nilai *fitness*-nya untuk dipilih sebagai populasi baru generasi selanjutnya. Populasi yang terpilih untuk generasi selanjutnya sebanyak populasi awal berdasarkan nilai *fitness* tertinggi yang diurutkan. Untuk mendapatkan nilai *fitness* yang lebih baik maka tahapan generasi akan terus diulang sampai banyaknya generasi dan ukuran populasi yang ditentukan.