

PENERAPAN METODE *FUZZY* TSUKAMOTO DALAM PENENTUAN JUMLAH PRODUKSI ROTI (STUDI KASUS: DWI JAYA BAKERY KUPANG)

Yunni A. Adoe¹, Kornelis Letelay¹, Emerensye S.Y. Pandie^{1*}

1. Jurusan Ilmu Komputer, Fakultas Sains dan Teknik, Universitas Nusa Cendana, Kupang, Indonesia

*penulis korespondensi: carolinepandie@gmail.com

ABSTRAK

Dwi Jaya Bakery merupakan sebuah usaha swasta yang bergerak dalam bidang produksi dan penjualan makanan khususnya roti. Ada 4 jenis roti yang di produksi yaitu: Roti Kecil Cokelat, Roti Kecil Mocca, Roti Besar Goreng Cokelat dan Roti Besar Abon. Pada penelitian ini penulis menerapkan metode *fuzzy* tsukamoto untuk menentukan jumlah produksi roti dengan 3 tahapan perhitungan yakni mendefinisikan variabel, inferensi dan penegasan (*defuzzifikasi*). Variabel *input* yang digunakan adalah permintaan dan persediaan serta variabel *output* jumlah produksi. Data produksi yang diperoleh dari industri Dwi Jaya Bakery adalah data pada bulan September 2018 - Februari 2019. Untuk pengujian data yang digunakan adalah sebanyak 28 data untuk setiap jenis rotinya yakni pada tanggal 1 sampai 28 februari 2019, yang sudah ditambah dengan persamaan transformasi dan diuji menggunakan pengujian *MAPE* (*Mean Absolut Percentage Error*) dimana akan dicari nilai kesalahan untuk data setiap jenis rotinya. Nilai kesalahan yang diperoleh dengan menerapkan metode pengujian *MAPE* yang sudah ditambah persamaan transformasi untuk jenis roti kecil cokelat adalah 1,936786%, roti kecil mocca 6,209643%, roti besar goreng cokelat 3,886071% dan roti besar abon 6,646429%.

Kata kunci: *fuzzy* tsukamoto, roti, *MAPE*, transformasi.

1. PENDAHULUAN

Roti merupakan sumber karbohidrat yang berbahan dasar utama tepung terigu dan air, yang difermentasikan dengan ragi, tetapi ada juga yang tidak menggunakan ragi. Roti dibuat bukan hanya untuk memenuhi kebutuhan konsumsi keluarga tetapi roti sudah diproduksi untuk dipasarkan. Produksi ini dilakukan oleh industri perdagangan yang bergerak di bidang makanan, mulai dari industri yang berskala rumahan hingga industri yang berskala besar. Dwi Jaya Bakery merupakan salah satu pabrik industri perdagangan yang bergerak di bidang produksi roti yang terletak di Jl. W.J Lalamentik, dan sudah ada sejak tahun 2001. Ada empat jenis roti yang di produksi di pabrik ini yaitu roti kecil cokelat, roti kecil mocca, roti besar goreng cokelat dan roti besar abon. Roti dengan ukuran kecil di jual dengan harga Rp1000 dan roti dengan ukuran besar yang di jual dengan harga Rp 2000. Pabrik ini mempunyai 18 pekerja yang bekerja setiap harinya. Dwi Jaya Bakery memproduksi

roti tanpa melihat jumlah permintaan dan jumlah persediaan yang ada, jika terjadi produksi berlebihan maka roti tersebut akan disimpan sebagai persediaan. Roti mempunyai masa konsumsi yang tidak bertahan lama sekitar 5-6 hari sehingga tidak dapat dikonsumsi setelah hari itu karena roti akan berjamur. Di sisi lain apabila terjadi kekurangan produksi roti maka pelanggan akan kecewa dan pabrik dirugikan. Dwi Jaya Bakery masih belum mampu menghasilkan jumlah produksi yang sesuai dengan jumlah permintaan dan persediaan yang ada sehingga bisa terjadi kelebihan dan kekurangan jumlah produksi roti.

Dwi Jaya Bakery membutuhkan sebuah sistem yang digunakan untuk menentukan jumlah produksi roti yang tepat sesuai dengan jumlah permintaan dan persediaan yang ada dengan menerapkan suatu metode. Ada banyak metode yang dapat digunakan untuk mengatasi masalah penambahan atau pengurangan jumlah produksi roti, salah satu metode yang dapat digunakan yaitu logika *fuzzy* yakni metode *Fuzzy Tsukamoto*.

Menurut Kusumadewi [1] metode Tsukamoto merupakan perluasan dari penalaran monoton. Pada metode Tsukamoto, setiap konsekuensi pada aturan yang berbentuk IF-Then harus dipresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Kelebihan dari metode ini yaitu lebih cepat dalam melakukan komputasi, lebih intuitif, diterima oleh banyak pihak, lebih cocok untuk masukan yang diterima dari manusia dan bukan mesin.

2. METODE

2.1 Logika Fuzzy

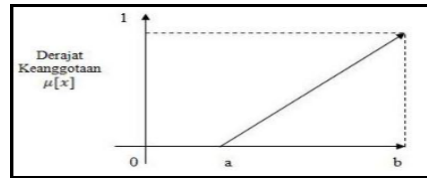
Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam ruang *output* [1]–[3]. Untuk sistem yang sangat rumit, penggunaan logika *fuzzy* adalah salah satu pemecahannya. Sistem tradisional dirancang untuk mengontrol keluaran tunggal yang berasal dari beberapa masukan yang tidak saling berhubungan. Penambahan masukan yang baru akan memperumit proses kontrol dan membutuhkan proses perhitungan kembali dari semua fungsi. Kebalikannya, penambahan masukan baru pada sistem *fuzzy*, yaitu sistem yang bekerja berdasarkan prinsip-prinsip logika *fuzzy*, hanya membutuhkan penambahan fungsi keanggotaan yang baru dan aturan-aturan yang berhubungan dengannya.

Beberapa fungsi yang biasa digunakan untuk mencari fungsi keanggotaan menurut Kusumadewi dan Purnomo [4]:

a. Representasi Linear

Pada representasi *linear*, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai sebuah garis lurus. Representasi kurva linear ini terbagi menjadi 2, yaitu representasi kurva linear naik dan representasi kurva linear turun dan masing-masing kurva memiliki fungsi keanggotaan yang berbeda.

Representasi *linear* naik merupakan kenaikan himpunan dimulai pada nilai *domain* yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai *domain* yang memiliki derajat keanggotaan lebih tinggi. Dapat dilihat pada gambar 2.1.



Gambar 2.1 Representasi *Linear Naik*

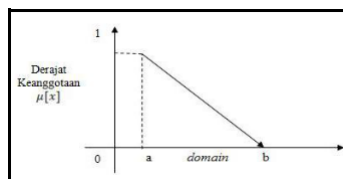
Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \dots\dots\dots(2.1)$$

Dengan :

- a = nilai *domain* yang mempunyai derajat keanggotaan nol (0)
- b = nilai *domain* yang mempunyai derajat keanggotaan satu (1)
- x = nilai *input* yang akan diubah ke dalam bilangan *fuzzy*

Representasi *linear* turun merupakan kebalikan dari representasi naik. Garis lurus dimulai dari nilai *domain* dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Dapat dilihat pada gambar 2.2.



Gambar 2.2 Representasi *Linear Turun*

Fungsi keanggotaan :

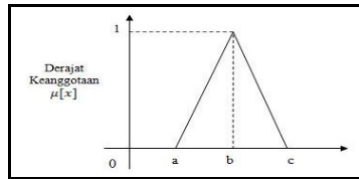
$$\mu[x] = \begin{cases} 0; & x \geq b \\ \frac{b-x}{b-a}; & a \leq x \leq b \\ 1; & x \leq a \end{cases} \dots\dots\dots(2.2)$$

Dengan :

- a= nilai *domain* yang mempunyai derajat keanggotaan satu (1)
- b = nilai *domain* yang mempunyai derajat keanggotaan nol (0)
- x = nilai *input* yang akan diubah ke dalam bilangan *fuzzy*

a. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (*linear*). Nilai-nilai disekitar b memiliki derajat keanggotaan turun cukup tajam (menjauhi 1). Dapat dilihat pada gambar 2.3.



Gambar 2.3 Kurva Segitiga

Fungsi keanggotaan :

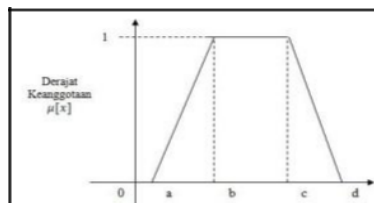
$$\mu[x] = \begin{cases} 0; & x \geq c \text{ atau } x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ \frac{c-x}{c-b}; & b \leq x \leq c \end{cases} \dots\dots\dots(2.3)$$

Dengan :

- a = nilai *domain* terkecil yang mempunyai derajat keanggotaan (0)
- b = nilai *domain* yang mempunyai derajat keanggotaan satu (1)
- c = nilai *domain* terbesar yang mempunyai derajat keanggotaan(0)
- x = nilai *input* yang akan diubah ke dalam bilangan *fuzzy*

b. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Dapat dilihat pada gambar 2.4.



Gambar 2.4 Kurva Trapesium

Fungsi keanggotaan :

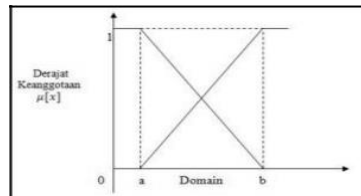
$$\mu[x] = \begin{cases} 0; & x \geq d \text{ atau } x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c}; & c \leq x \leq d \end{cases} \dots\dots\dots(2.4)$$

Dengan :

- a = nilai *domain* terkecil yang mempunyai derajat keanggotaan nol (0)
- b = nilai *domain* terkecil yang mempunyai derajat keanggotaan satu (1)
- c = nilai *domain* terbesar yang mempunyai derajat keanggotaan satu (1)
- d = nilai *domain* terbesar yang mempunyai derajat keanggotaan nol (0)
- x = nilai *input* yang akan diubah ke dalam bilangan *fuzzy*

c. Representasi Kurva Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Himpunan *fuzzy* "bahu", bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar. Dapat dilihat pada gambar 2.5.



Gambar 2.5 Kurva Bahu

2.2 Fuzzy Tsukamoto

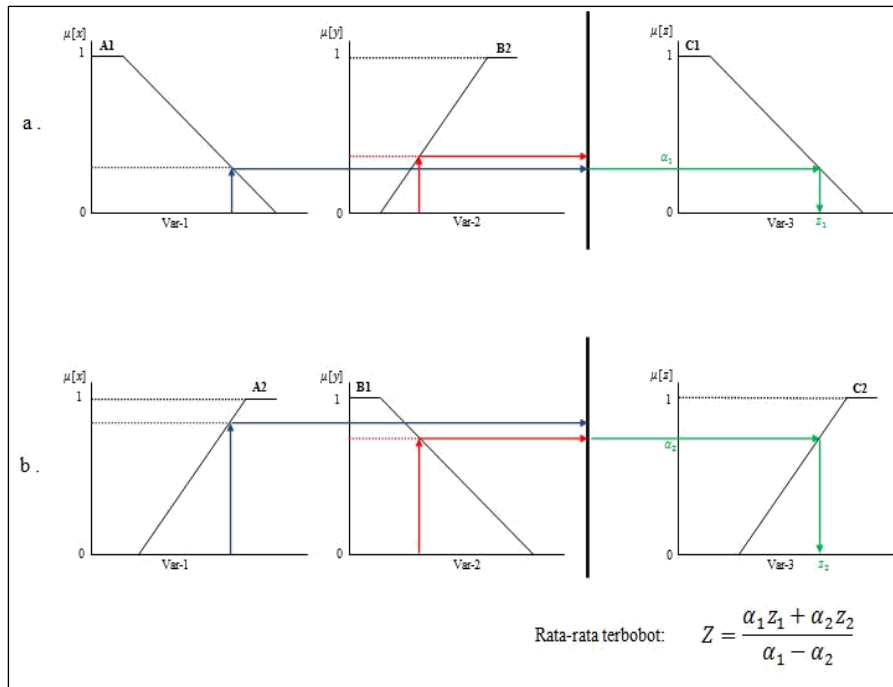
Menurut Kusumadewi [4], Metode Tsukamoto merupakan perluasan dari penalaran monoton. Pada metode Tsukamoto, Setiap konsekuen pada aturan yang berbentuk IF-THEN harus dipresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

Misalkan ada 2 variabel input, Var-1 (x) dan Var-2 (y), serta 1 variabel output, Var-3(z), dimana Var-1 terbagi atas 2 himpunan yaitu A_1 dan A_2 terbagi atas 2 himpunan B_1 dan B_2 , Var-3 juga terbagi atas 2 himpunan yaitu C_1 dan C_2 (C_1 dan C_2 HARUS MONOTON). Ada 2 aturan yang digunakan, yaitu :

[R1] IF (x is A_1) and (y is B_2) THEN (z is C_1)

[R2] IF (x is A_2) and (y is B_1) THEN (z is C_2)

Alur inferensi untuk mendapatkan satu nilai *crisp* z seperti terlihat pada Gambar 2.6.



Gambar 2.6 Inferensi dengan menggunakan Metode Tsukamoto

Dalam prosesnya, metode fuzzy tsukamoto memiliki beberapa tahapan, yaitu:

1. *Fuzzifikasi*

Fuzzifikasi adalah proses untuk mengubah masukan sistem yang mempunyai nilai tegas atau crisp menjadi himpunan fuzzy dan menentukan derajat keanggotaannya di dalam himpunan fuzzy.

2. Inferensi

Proses untuk membentuk rule yang akan digunakan dalam bentuk *IF –THEN* yang tersimpan dalam basis keanggotaan *fuzzy* dan mengubah masukan fuzzy menjadi keluaran *fuzzy* dengan cara *fuzzifikasi* tiap rule (*IF-THEN Rules*) yang telah ditetapkan.

Tabel 2.1 Aturan *Fuzzy*

No	<i>If</i>		<i>Then</i>	<i>Rule</i>
	Permintaan	Persediaan	Jumlah Produksi	
1	Rendah	Rendah	Rendah	R1
2	Rendah	Sedang	Rendah	R2
3	Rendah	Tinggi	Rendah	R3
4	Sedang	Rendah	Sedang	R4
5	Sedang	Sedang	Sedang	R5
6	Sedang	Tinggi	Sedang	R6
7	Tinggi	Rendah	Tinggi	R7
8	Tinggi	Sedang	Tinggi	R8
9	Tinggi	Tinggi	Tinggi	R9

3. *Defuzzifikasi* Mengubah keluaran *fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas atau *crisp*. Hasil akhir diperoleh dengan menggunakan persamaan rata-rata pembobotan menggunakan metode rata-rata *weight average*.

2.3 Perencanaan Produksi

Perencanaan produksi adalah aktivitas untuk menetapkan produk yang diproduksi, jumlah yang dibutuhkan, kapan produk tersebut harus selesai dan sumber-sumber yang dibutuhkan. Pengendalian produksi adalah aktivitas yang menetapkan kemampuan sumber-sumber yang digunakan dalam memenuhi rencana, kemampuan produksi berjalan sesuai rencana, melakukan perbaikan rencana. Tujuan utamanya adalah memaksimalkan pelayanan bagi konsumen, meminimumkan investasi pada persediaan, perencanaan kapasitas, pengesahan produksi dan pengesahan pengendalian produksi, persediaan dan kapasitas, penyimpanan dan pergerakan material, peralatan, routing dan proses planning, dan sebagainya [5], [6].

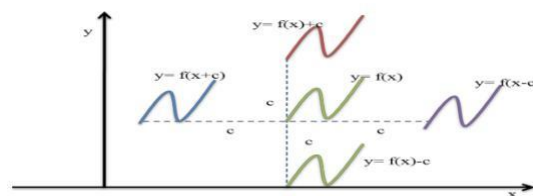
Penentuan jumlah produksi yang tepat, merupakan suatu hal yang harus dimiliki oleh sebuah perusahaan. Penentuan produksi yang kurang tepat akan menyebabkan perusahaan mengalami kerugian karena bisa jadi perusahaan terlalu berlebihan dalam memproduksi atau terlalu sedikit dalam memproduksi sehingga tidak sesuai dengan permintaan pelanggan [7].

2.4 Transformasi

Transformasi adalah salah satu cara untuk memanipulasi yang paling umum. Fungsi transformasi terdiri dari mengalikan, membagi, menambah, atau mengurangi konstanta pada *input* atau *output*. Menambahkan konstanta untuk fungsi ini dipanggil dengan translasi dan mengalikan dengan fungsi konstanta dengan dilatasi.

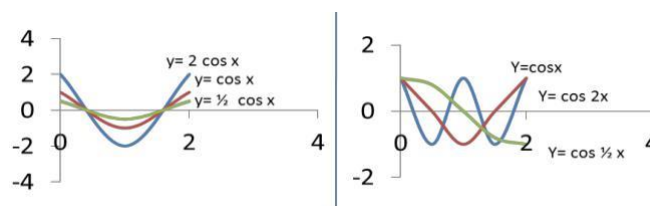
Berikut jenis-jenis transformasi yang dapat dilakukan antara lain:

- a. Translasi (Pergeseran)



Gambar 2.7 Grafik Translasi

- b. Dilatasi (Peregangan)



Gambar 2.8 Grafik Dilatasi

Pada penelitian ini persamaan dilatasi digunakan untuk merapatkan dan meregangkan jumlah roti yang harus diproduksi sedangkan persamaan translasi digunakan untuk menghitung pergeseran sejauh konstantanya, sehingga persamaan yang digunakan adalah:

1. Peregangan Vertikal

$$y = cf(x) \dots\dots\dots(2.5)$$

dengan:

y = jumlah produksi tertinggi baru

$f(x)$ = jumlah roti yang harus diproduksi yakni permintaan dikurangi persediaan

c = hasil bagi jumlah produksi tertinggi dan sedang

2. Perapatan Vertikal

$$y = (1/c)f(x) \quad \dots\dots\dots (2.6)$$

dengan:

y = jumlah produksi terendah baru

$f(x)$ = jumlah roti yang harus diproduksi yakni permintaan dikurangi persediaan

c = hasil bagi jumlah produksi sedang dan jumlah produksi rendah

3. Pergeseran Vertikal

$$y = f(x) + k \quad \dots\dots\dots (2.7)$$

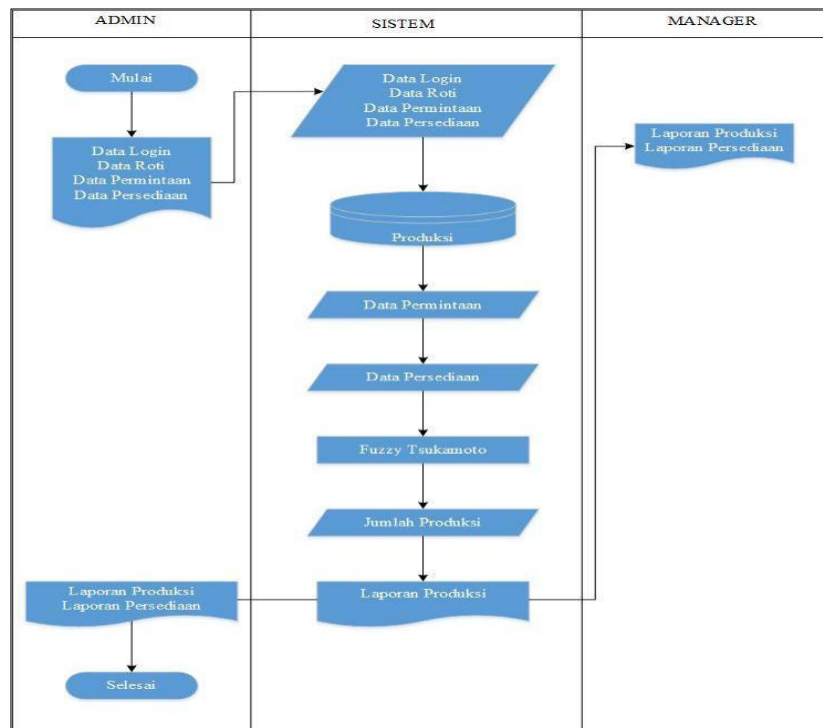
dengan:

$f(x)$ = jumlah produksi tertinggi baru, sedang dan terendah

k = konstanta yang menjadi besar pergeseran

2.5 Bagan Alir (Flowchartl)

Menurut Jogiyanto (2005), bagan alir (*flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada didalam sistem. Gambaran *flowchart* dari sistem yang akan dibuat dapat dilihat pada Gambar 2.9



Gambar 2.9 Flowchart Sistem Yang Diusulkan

3. HASIL DAN PEMBAHASAN

3.1 Hasil Perancangan Sistem

Hasil penerapan metode *fuzzy tsukamoto* untuk menentukan jumlah produksi roti merupakan antarmuka yang memudahkan pengguna saat ingin menggunakan sistem yang ada. Implementasi antarmuka berupa tampilan sistem yang memuat informasi kepada pengguna dengan hak akses sistem kepada pengguna yakni *admin* yang merupakan manajer atau pemilik dari Dwi Jaya Bakery.

3.1.1 Tampilan Halaman Utama

Tampilan Halaman Utama menampilkan halaman depan dari sistem yang dibangun dan pada menu ini juga terdapat tombol login yang dapat digunakan untuk mengakses menu login, seperti yang dapat digambarkan pada gambar 3.1.



Gambar 3.1 Tampilan Halaman Utama

3.1.2 Tampilan Menu *Dashboard*

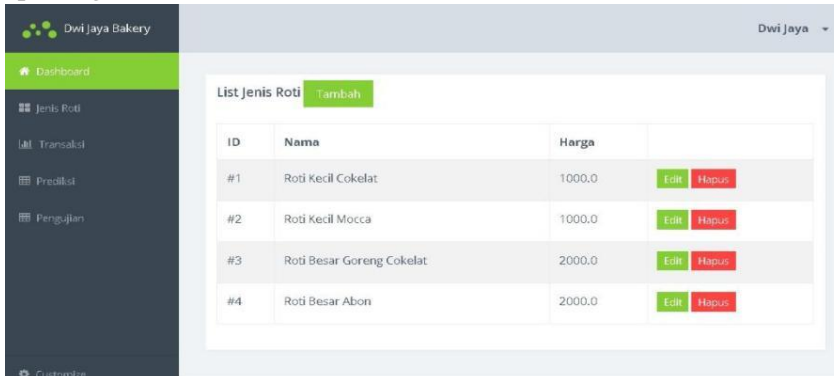
Tampilan menu *dashboard* menyajikan data maksimum, tengah dan minimum untuk setiap jenis roti yang ada dan juga rangkuman 10 hari terakhir dari permintaan, persediaan dan produksi setiap jenis roti. Dapat di lihat pada gambar 3.3.

Max, Mid, Min									
Jenis Roti	Max Prod	Mid Prod	Min Prod	Max Perm	Mid Perm	Min Perm	Max Pers	Mid Pers	Min Pers
Roti Kecil Cokelat	9500	7750.0	6000	9488	7747.0	6006	845	422.5	0
Roti Kecil Mocca	3300	2150.0	1000	3300	2155.0	1010	605	305.0	5
Roti Besar Goreng Cokelat	7500	5100.0	2700	7288	5066.0	2844	616	308.0	0
Roti Besar Abon	3500	2100.0	700	3470	2095.0	720	650	339.0	28

Gambar 3.3 Tampilan Menu *Dashboard*

3.1.3 Tampilan Menu Jenis Roti

Menu jenis roti adalah menu yang menampilkan daftar jenis roti beserta harganya, dapat dilihat pada gambar 3.4.



ID	Nama	Harga	
#1	Roti Kecil Cokelat	1000.0	Edit Hapus
#2	Roti Kecil Mocca	1000.0	Edit Hapus
#3	Roti Besar Goreng Cokelat	2000.0	Edit Hapus
#4	Roti Besar Abon	2000.0	Edit Hapus

Gambar 3.4 Tampilan Menu Jenis Roti

3.1.4 Tampilan Menu Transaksi

Tampilan menu transaksi adalah menu yang menyajikan data produksi berupa tanggal, jenis roti, permintaan, persediaan serta jumlah produksi setiap hari. Data yang ditampilkan adalah data produksi selama 6 bulan yaitu dari bulan September 2018-Februari 2019 yang disajikan dalam bentuk tabel, seperti yang dapat dilihat dalam gambar 3.5.




Tanggal	Jenis Roti	Permintaan	Persediaan	Produksi	
28-02-2019	Roti Besar Abon	2956	89	3000	Edit Hapus
27-02-2019	Roti Besar Abon	3003	92	3000	Edit Hapus
26-02-2019	Roti Besar Abon	2078	170	2000	Edit Hapus
25-02-2019	Roti Besar Abon	1738	108	1800	Edit Hapus
24-02-2019	Roti Besar Abon	1873	81	1900	Edit Hapus
23-02-2019	Roti Besar Abon	2666	147	2600	Edit Hapus

Gambar 3.5 Tampilan Menu Transaksi

3.1.5 Tampilan Menu Prediksi

Menu prediksi merupakan menu untuk melakukan perhitungan jumlah produksi roti menggunakan metode *fuzzy tsukamoto* dengan memilih jenis roti yang ingin dihitung, tanggal, permintaan, dan produksi selanjutnya akses tombol *submit* maka sistem secara otomatis akan menentukan jumlah produksi. Dapat dilihat pada gambar 3.6.



Jenis Roti
Roti Kecil Cokelat

Tanggal
mm/dd/yyyy

Permintaan

Persediaan

Submit

Gambar 3.6 Tampilan Menu Prediksi

Pada menu ini sistem akan secara otomatis menghitung jumlah produksi menggunakan metode *fuzzy tsukamoto* yang terdiri dari 3 langkah yakni mendefinisikan variable (*fuzzifikasi*), inferensi dan penegasan (*defuzzifikasi*). Gambar 3.7 merupakan *Source code* langkah pertama *fuzzy tsukamoto* yaitu mendefinisikan variabel.

```
1 def build_rules(data):
2     data_perm = [row for row in data[0]]
3     data_pers = [row for row in data[1]]
4     data_prod = [row for row in data[2]]
5     membership_perm = build_criteria_membership(data_perm)
6     membership_pers = build_criteria_membership(data_pers)
7     inv_membership_prod = build_inverse_membership(data_prod)
8     result = []
9     for r in RULES:
10        perm, pers, prod = r
11        rule = Rule(
12            persediaan=RuleAntecedent(level=pers, label="persediaan", membership=membership_pers),
13            permintaan=RuleAntecedent(level=perm, label="permintaan", membership=membership_perm),
14            produksi=RuleConsequent(level=prod, label="produksi", inv_membership=inv_membership_prod)
15        )
16        result.append(rule)
17    return result
```

Gambar 3.7 *Source code* Mendefinisikan Variabel

Berdasarkan gambar 3.7 dapat dijelaskan baris 1-4 merupakan persamaan yang digunakan untuk mengambil nilai dari permintaan, persediaan dan produksi yang telah di input. Selanjutnya pada baris 5-7 akan dibuat fungsi keanggotaan/himpunan *fuzzy* dari nilai permintaan, persediaan dan produksi yang telah diperoleh. Kemudian akan dilakukan iterasi pada tiap rule seperti pada baris 8-Langkah berikutnya setelah mendefinisikan variable adalah inferensi. Berikut merupakan *Source code* inferensi yang dapat dilihat pada gambar 3.8.

```
1 def _call_(self, x: RuleInput):
2     mem_vals = [ self.permintaan.membership(self.permintaan.level, x.permintaan),
3                 self.persediaan.membership(self.persediaan.level, x.persediaan) ]
4     min_x = min(mem_vals)
5     z = self.produksi.inv_membership(self.produksi.level, min_x, x.permintaan, x.persediaan)
6
7     return RuleResult(input=x,
8                       memberships_value=mem_vals,
9                       target_value=min_x,
10                      target_inverse=z,
11                      rule=self)
```

Gambar 3.8 *Source code* Inferensi

Gambar 3.8 merupakan *source code* langkah kedua yakni inferensi dimana akan ditentukan himpunan *fuzzy* rendah, sedang atau tinggi pada baris 1-3. Kemudian pada baris ke-4 akan di cari nilai minimum dari himpunan *fuzzy* yang telah di peroleh, lalu pada baris 5 sampai selesai akan di cari nilai z yang kemudian akan digunakan pada proses *defuzzifikasi*. Langkah terakhir penegasan (*defuzzifikasi*). Berikut adalah *Source code* dari langkah terakhir yang dapat dilihat dalam gambar 3.9.

```
1 def _defuzz(self, rows):
2
3     nom = [ r.target_value * r.target_inverse for r in rows ]
4     den = [ r.target_value for r in rows ]
5     return sum(nom) / sum(den)
```

Gambar 3.9 *Source code* Defuzzifikasi

Berdasarkan gambar 3.9 akan ditentukan nilai *defuzzifikasi* dengan cara menghitung rata-rata dari nilai z dengan nilai \min yang telah diperoleh pada langkah 2 seperti pada baris 3-5.

3.2 Hasil Pengujian Sistem

Pengujian sistem digunakan pada penelitian ini adalah pengujian *MAPE* (*Mean Absolute Percentage Error*). Berdasarkan hasil pengujian *MAPE* yang diperoleh setelah ditambahkan persamaan transformasi mendapatkan hasil yang lebih optimal dibanding dengan sebelum ditambahkan persamaan transformasi. Hal ini dibuktikan pada hasil pengujian dengan menggunakan data yang sama yakni data tanggal 1-28 februari 2019, hasil pengujian yang diperoleh dengan menggunakan metode *MAPE* untuk hasil uji sistem tanpa persamaan transformasi untuk roti kecil coklat adalah 4,923571, roti kecil mocca 11,51214%, roti besar goreng coklat 7,185714% dan roti besar abon 8,636071% lebih besar nilai kesalahannya daripada dengan tambahan persamaan transformasi dimana roti kecil coklat mendapat nilai kesalahan 1,936786%, roti kecil mocca 6,209643%, roti besar goreng coklat 3,886071% dan roti besar abon 6,646429%.

4. SIMPULAN

Berdasarkan hasil penelitian penerapan metode *fuzzy tsukamoto* untuk penentuan jumlah produksi roti pada industri Dwi Jaya Bakery maka dapat disimpulkan bahwa aplikasi Penerapan metode fuzzy Tsukamoto dalam penentuan jumlah produksi roti telah dibuat sesuai perancangan dan dapat digunakan dalam merekomendasikan penentuan jumlah produksi roti. Aplikasi ini juga memiliki kinerja sistem yang mampu berjalan sesuai dengan kebutuhan fungsional. Hal ini dibuktikan dengan hasil pengujian Black Box yang 100% sukses. Berdasarkan hasil pengujian *MAPE* (*Mean Absolute Percentage Error*) dengan tambahan persamaan transformasi terhadap 28 data uji pada bulan Februari 2019 untuk 4 jenis roti yakni roti kecil coklat, roti kecil mocca, roti besar goreng coklat dan roti besar abon. 4 jenis roti tersebut mendapatkan nilai presentase error yang rendah yakni di bawah 10%. Untuk jenis roti kecil coklat mendapatkan presentase error sebesar 1,936786%, roti kecil mocca mendapatkan presentase error sebesar 6,209643%, roti besar goreng coklat mendapatkan presentase error sebesar 3,886071% dan roti besar abon mendapatkan presentase eror sebesar 6,646429%.

DAFTAR PUSTAKA

- [1] S. Kusumadesi, *Aplikasi Logika Fuzzy Fungsi Keanggotaan*. Yogyakarta: Graha Ilmu, 2010.
- [2] C. Kahraman, S. C. Onar, and B. Oztaysi, "Fuzzy Multicriteria Decision-Making: A Literature Review," *International Journal of Computational Intelligence Systems*, vol. 8, no. 4, pp. 637–666, Jul. 2015, doi: 10.1080/18756891.2015.1046325.

- [3] G. Prabakaran, D. Vaithyanathan, and M. Ganesan, "Fuzzy decision support system for improving the crop productivity and efficient use of fertilizers," *Computers and Electronics in Agriculture*, vol. 150, pp. 88–97, Jul. 2018, doi: 10.1016/j.compag.2018.03.030.
- [4] S. Kusumadesi and H. Pernomo, *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu, 2004.
- [5] H. Kusuma, *Manajemen produksi: perencanaan dan pengendalian produksi*. Yogyakarta: Penerbit ANDI, 2002.
- [6] A. Eunike, N. W. Setyanto, R. Yuniarti, I. Hamdala, R. P. Lukodono, and A. A. Fanani, *Perencanaan Produksi dan Pengendalian Persediaan: Edisi Revisi*. Universitas Brawijaya Press, 2021. [Online]. Available: <https://books.google.co.id/books?id=8AJWEAAAQBAJ>
- [7] A. H. Nasution and Y. Prasetyawan, *Perencanaan dan pengendalian produksi*. Yogyakarta: Graha Ilmu, 2008.