TECHNOVATAR Jurnal Teknologi, Industri dan Informasi
Vol. 1, No. 1, Oktober 2023, pp. 54 – 67
Open Access: https://journal.awatarapublisher.com/index.php/technovatar
ISSN: 3031-8874, **DOI**: 10.61434/technovatar.v1i1.60

DETEKSI STRES MANUSIA MELALUI ANALISIS TIDUR DENGAN METODE FUZZY

Zain Muzadid Zamzani[1], M. Ryan Nurdiansyah N.A[2], Baktiar Yudha Yana[3]
[1,2,3] Program Studi Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

e-mail: 21081010174@student.upnjatim.ac.id[1], 21081010321@student.upnjatim.ac.id[2], 21081010324@student.upnjatim.ac.id[3]

INFO ARTIKEL
Sejarah Artikel:
Diterima, 21 Juni 2023
Direvisi, 25 September 2023
Disetujui, 5 Oktober 2023

KEYWORDS
Stress detection
Sleep analysis
Mamdani fuzzy method
Tsukamoto's fuzzy method
Level of accuracy

ABSTRACT

Detection and measurement of stress levels in individuals is an important aspect in the fields of health and psychology. One emerging approach is through sleep analysis, which can provide insight into a person's stress state. The aim of this research is to identify a fuzzy method that can provide a more accurate estimate of stress levels through sleep analysis.

In this research, two fuzzy methods, namely the Mamdani Method and the Tsukamoto Method, were applied to analyze sleep patterns and obtain estimates of stress levels. The Mamdani Method and Tsukamoto Method use fuzzy rules to relate sleep variables to estimated stress levels.

The research results show that the Fuzzy Mamdani Method has an accuracy rate of 84.920635% in estimating stress levels through sleep analysis. Meanwhile, the Fuzzy Tsukamoto Method obtained an accuracy rate of 83.174603%. Thus, the Fuzzy Mamdani Method is significantly superior in providing more accurate estimates compared to the Fuzzy Tsukamoto Method. This discovery makes an important contribution to the development of stress detection methods through sleep analysis, which can be used in health and psychological research to help identify and treat stress problems in individuals.

This work is licensed under a Creative Commons Attribution 4.0 International License.

KATA KUNCI
Deteksi stress
Analisis tidur
Metode fuzzy Mamdani
Metode fuzzy Tsukamoto
Tingkat akurasi

CORRESPONDING AUTHOR
Anggraini Puspita Sari
Program Studi Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia
anggraini.puspita.if@upnjatim.ac.id

ABSTRAK

Deteksi dan pengukuran tingkat stres pada individu merupakan aspek penting dalam bidang kesehatan dan psikologi. Salah satu pendekatan yang sedang berkembang adalah melalui analisis tidur, yang dapat memberikan wawasan tentang kondisi stres seseorang. Tujuan dari penelitian ini adalah untuk mengidentifikasi metode fuzzy yang dapat memberikan estimasi tingkat stres yang lebih akurat melalui analisis tidur.

Dalam penelitian ini, dua metode fuzzy, yaitu Metode Mamdani dan Metode Tsukamoto, diterapkan untuk menganalisis pola tidur dan memperoleh estimasi tingkat stres. Metode Mamdani dan Metode Tsukamoto menggunakan aturan-aturan fuzzy untuk menghubungkan variabel-variabel tidur dengan tingkat stres yang diestimasikan.

Hasil penelitian menunjukkan bahwa Metode Fuzzy Mamdani memiliki tingkat akurasi sebesar 84.920635% dalam mengestimasi tingkat stres melalui analisis tidur. Sementara itu, Metode Fuzzy Tsukamoto memperoleh tingkat akurasi sebesar 83.174603%. Dengan demikian, Metode Fuzzy Mamdani secara signifikan lebih unggul dalam memberikan estimasi yang lebih akurat dibandingkan dengan Metode Fuzzy Tsukamoto. Penemuan ini memberikan kontribusi penting dalam pengembangan metode deteksi stres melalui analisis tidur, yang dapat digunakan dalam bidang kesehatan dan penelitian psikologis untuk membantu mengidentifikasi dan mengatasi masalah stres pada individu.

PENDAHULUAN

Stres merupakan fenomena yang dialami setiap orang. Dalam psikologi,Stres adalah perasaan tertekan dan ketegangan mental. Tingkat stres yang rendah bisadiinginkan, bermanfaat dan

bahkan sehat. Stres dapat memberikan efek positif yaitu dapatpeningkatan kinerja yang menguntungkan. Stres positif dianggap sebagai faktor penting untukmotivasi, adaptasi, dan respons terhadap lingkungan. Namun, tingkat stresTingkat tinggi dapat menyebabkan masalah biologis, psikologis dan sosial dan bahkan menyebabkan kerusakan serius seseorang. Stres bisa berasal dari faktor eksternal di lingkungan, ataudisebabkan oleh persepsi internal individu.[16]

Tidur adalah keadaan istirahat yang dibutuhkan manusia secara teraturTidur ini ditandai dengan berkurangnya gerakan tubuh dan berkurangnya kewaspadaan lingkungan. Tidur memiliki dampak besar pada kesehatan fisik.mental, emosional dan sistem kekebalan tubuh. Orang yang kurang tidur cenderung lebih mudah penyakit, seperti serangan jantung, anemia, dan tekanan darah tinggi.[12]

Salah satu metode yang digunakan dalam analisis tidur untuk deteksi stres adalah metode fuzzy. Metode fuzzy memungkinkan untuk menangani data yang ambigu atau tidak pasti dengan memperkenalkan tingkat keanggotaan dalam variabel yang akan dianalisis. Dalam konteks deteksi stres melalui analisis tidur, metode fuzzy dapat digunakan untuk menginterpretasikan data tidur yang kompleks dan menghasilkan tingkat stres terkait.

Metode fuzzy dapat mengintegrasikan beberapa variabel tidur seperti durasi tidur, terbangun di malam hari, frekuensi perubahan posisi, dan pola pernapasan untuk menghasilkan nilai fuzzy yang merepresentasikan tingkat stres. Menggunakan aturan fuzzy yang telah ditentukan sebelumnya, tingkat stres dapat diklasifikasikan ke dalam kategori seperti rendah, sedang atau tinggi.

METODE PENELITIAN

Metode penelitian ini bertujuan untuk mendeteksi tingkat stres manusia melalui analisis tidur dengan menggunakan pendekatan fuzzy. Langkah-langkah yang terdiri dalam metode ini adalah sebagai berikut. Pertama, kriteria yang relevan untuk mendeteksi tingkat stres manusia melalui analisis tidur ditentukan dengan variable seperti Tingkat suara mendengkur pasien, Tingkat pernapasan pasien, Suhu tubuh pasien, Gerakan anggota tubuh pasien, Kadar oksigen dalam darah pasien, Gerakan mata pasien saat tidur, Durasi tidur pasien dalam jam,mDetak jantung pasien, Tingkat stres yang sebenarnya pada pasien (0-4).

Selain itu, informasi istirahat yang signifikan dikumpulkan dari sumber yang dapat diandalkan. Informasi ini pada saat itu dianalisis dan disiapkan untuk membuat demonstrasi halus yang sesuai dengan tingkat peregangan pengguna berdasarkan faktor-faktor di atas.

Sistem inferensi lembut digunakan untuk mengasosiasikan input klien, menjadi informasi istirahat spesifik, dengan hasil yang diperlukan, menjadi penemuan tingkat dorongan manusia tertentu. Kerangka kerja induksi ini menggunakan aturan halus yang muncul dari pemeriksaan data untuk menghitung tingkat kesesuaian antara faktor pemeriksaan istirahat dan tingkat peregangan.

Strategi ini kemudian dicoba menggunakan uji dataset yang terdiri dari beberapa faktor di atas. Munculnya antisipasi tingkat dorongan manusia dari strategi berbulu Mamdani dan berbulu halus Tsukamoto akan memberikan informasi dan tingkat akurasi yang sama.

Penelitian ini diharapkan dapat mengidentifikasi dorongan manusia melalui penyelidikan tidur dengan menggabungkan pendekatan halus dan membantu klien dalam memahami dan memantau tingkat dorongan mereka berdasarkan karakteristik tidur.

Teknik Pengumpulan Data

Untuk mengumpulkan data dalam rangka Deteksi Stres Manusia melalui Analisis Tidur dengan metode fuzzy yaitu Informasi tentang analisis tidur, seperti Tingkat suara mendengkur pasien, Tingkat pernapasan pasien, Suhu tubuh pasien, Gerakan anggota tubuh pasien, Kadar oksigen dalam

darah pasien, Gerakan mata pasien saat tidur, Durasi tidur pasien dalam jam,mDetak jantung pasien, Tingkat stres yang sebenarnya pada pasien (0-4). Data ini dapat memberikan wawasan tentang analisis tidur yang dapat digunakan dalam proses deteksi stress manusia. Data ini diambil dari keaggle.

Validasi Data

Validitas Data dalam Deteksi Stres Manusia melalui Analisis Tidur dengan Metode Fuzzy. Dalam ini, validitas data juga menjadi penting untuk memastikan akurasi deteksi. Berikut adalah beberapa faktor yang dapat mempengaruhi validitas data dalam metode ini:

1. Sumber Data:
   Sangat penting untuk mendapatkan informasi dari sumber yang dapat dipercaya dan bertanggung jawab
2. Akurasi Data:
   Informasi yang dikumpulkan harus tepat dan bebas dari kesalahan atau kecenderungan. Konfirmasi dan persetujuan atas informasi yang diperoleh sebelumnya untuk digunakan akan memberikan bantuan yang menjamin keakuratan data yang digunakan dalam pemeriksaan.
3. Ukuran Sampel yang Representatif:
   Jika informasi dikumpulkan melalui penelitian atau inspeksi, penting untuk memiliki estimasi pengujian agen yang cukup besar dan efektif. Ini akan memberikan bantuan untuk mengurangi kecenderungan dan meningkatkan generalisasi dari hasil pemeriksaan.
4. Konsistensi Data:
   Pastikan informasi yang dikumpulkan stabil dan tidak bentrok. Dalam hal terdapat kejanggalan dalam informasi selebihnya, sangat mendasar untuk dilakukan verifikasi ulang atau pengecekan keabsahan data sebelum digunakan dalam investigasi.
5. Pemilihan Variabel yang Relevan:
   Sangat penting untuk memilih faktor istirahat yang relevan dalam pemeriksaan. Pastikan faktor yang digunakan benar-benar relevan dan dapat memberikan data penting dalam mendeteksi tingkat tekanan manusia.

Dengan memperhatikan variabel-variabel di atas dan melakukan persetujuan yang tepat, dapat meningkatkan legitimasi informasi dalam mengenali dorongan manusia melalui pemeriksaan istirahat dengan menggunakan strategi lembut. Keabsahan informasi yang baik akan menawarkan bantuan yang menjamin kebenaran dan ketepatan harapan yang berasal dari tingkat manusia berdasarkan penyelidikan yang lain.

Teknik Analisa Data

Dalam konteks deteksi stres manusia melalui analisis tidur dengan metode fuzzy Mamdani dan tsukamoto, terdapat beberapa teknik analisis data yang dapat digunakan. Berikut adalah beberapa teknik yang digunakan dalam analisis data untuk deteksi stres manusia melalui analisis tidur dengan metode fuzzy Mamdani dan tsukamoto:

1. Fungsi Keanggotaan:
   Teknik ini melibatkan pembentukan fungsi keanggotaan (membership function) yang digunakan untuk menggambarkan tingkat keanggotaan suatu variabel dalam himpunan fuzzy. Fungsi keanggotaan ini dapat digunakan untuk menggambarkan tingkat stres manusia berdasarkan karakteristik tidur, seperti Tingkat suara mendengkur pasien, Tingkat pernapasan pasien, Suhu tubuh pasien, Gerakan anggota tubuh pasien, Kadar oksigen dalam darah pasien, Gerakan mata pasien saat tidur, Durasi tidur pasien dalam jam,mDetak jantung pasien, Tingkat stres yang sebenarnya pada pasien (0-4).

2. Inferensi Fuzzy:
   Inferensi fuzzy adalah teknik yang digunakan untuk mengambil keputusan atau menghasilkan tingkat stres manusia berdasarkan input yang diberikan dan aturan-aturan fuzzy yang telah ditentukan sebelumnya. Inferensi fuzzy memanfaatkan logika fuzzy untuk menggabungkan informasi yang ada dan menghasilkan tingkat stres manusia sebagai output.
3. Operator Logika Fuzzy:
   Operator logika fuzzy, seperti operator AND, OR, dan NOT, digunakan untuk menggabungkan nilai keanggotaan dari berbagai variabel yang digunakan dalam analisis tidur. Operator ini memungkinkan penggabungan informasi dari berbagai aspek tidur, seperti Tingkat suara mendengkur pasien, Tingkat pernapasan pasien, Suhu tubuh pasien, Gerakan anggota tubuh pasien, Kadar oksigen dalam darah pasien, Gerakan mata pasien saat tidur, Durasi tidur pasien dalam jam,mDetak jantung pasien, Tingkat stres yang sebenarnya pada pasien (0-4). sehingga dapat menghasilkan tingkat stres manusia yang lebih komprehensif.
4. Defuzzifikasi:
   Defuzzifikasi adalah tahap dalam metode fuzzy di mana output fuzzy yang diperoleh dari inferensi fuzzy dikonversi menjadi nilai tegas atau keputusan yang konkret. Teknik defuzzifikasi yang umum digunakan adalah metode centroid, di mana nilai tegas dihitung berdasarkan pusat massa dari himpunan fuzzy output yang diperoleh.
5. Evaluasi dan Validasi:
   Setelah mendapatkan output tingkat stres manusia, data dapat dievaluasi dan divalidasi untuk meprediksi tingkat akurasi prediksi.

Dengan menerapkan teknik-teknik analisis data tersebut, dapat dihasilkan model fuzzy yang dapat mendeteksi tingkat stres manusia berdasarkan analisis tidur. Penting untuk melakukan evaluasi dan validasi yang tepat guna memastikan keakuratan dan validitas prediksi tingkat stres manusia yang dihasilkan melalui metode fuzzy ini.

DISKUSI DAN PEMBAHASAN

Beberapa penelitian telah melakukan pengujian dan implementasi metode fuzzy untuk mendeteksi stress melalui analisis tidur dengan hasil yang diinginkan. Hasil - hasil penelitian ini melibatkan penggunaan teknik-teknik fuzzy logic untuk memproses data dan menghasilkan deteksi yang lebih personal dan sesuai dengan preferensi individu. Beberapa hasil penelitian tersebut meliputi:

1. Mendefinisikan Variabel:

```
# Membuat variabel fuzzy
snoring_rate = ctrl.Antecedent(np.arange(45, 101, 1), 'snoring_rate')
respiration_rate = ctrl.Antecedent(np.arange(16, 31, 1), 'respiration_rate')
body_temperature = ctrl.Antecedent(np.arange(85, 100, 1), 'body_temperature')
limb_movement = ctrl.Antecedent(np.arange(4, 20, 1), 'limb_movement')
blood_oxygen = ctrl.Antecedent(np.arange(82, 98, 1), 'blood_oxygen')
eye_movement = ctrl.Antecedent(np.arange(60, 106, 1), 'eye_movement')
sleeping_hours = ctrl.Antecedent(np.arange(0, 10, 1), 'sleeping_hours')
heart_rate = ctrl.Antecedent(np.arange(50, 86, 1), 'heart_rate')
stress_level = ctrl.Consequent(np.arange(0, 11, 1), 'stress_level')
```

   - snoring_rate: Variabel ini menggambarkan tingkat deru saat tidur, dengan rentang nilai antara 45 hingga 100. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "snoring_rate".
   - respiration_rate: Variabel ini menggambarkan tingkat frekuensi pernapasan, dengan rentang nilai antara 16 hingga 30. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "respiration_rate".

- body_temperature: Variabel ini menggambarkan suhu tubuh, dengan rentang nilai antara 85 hingga 99. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "body_temperature".
- limb_movement: Variabel ini menggambarkan tingkat gerakan anggota tubuh, dengan rentang nilai antara 4 hingga 19. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "limb_movement".
- blood_oxygen: Variabel ini menggambarkan tingkat oksigen dalam darah, dengan rentang nilai antara 82 hingga 97. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "blood_oxygen".
- eye_movement: Variabel ini menggambarkan tingkat gerakan mata saat tidur, dengan rentang nilai antara 60 hingga 105. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "eye_movement".
- sleeping_hours: Variabel ini menggambarkan durasi tidur dalam jam, dengan rentang nilai antara 0 hingga 9. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "sleeping_hours".
- heart_rate: Variabel ini menggambarkan detak jantung, dengan rentang nilai antara 50 hingga 85. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "heart_rate".
- stress_level: Variabel ini adalah variabel konsekuen yang menggambarkan tingkat stres. Rentang nilai variabel ini adalah antara 0 hingga 10. Rentang nilai ini dibagi menjadi nilai-nilai diskrit dengan inkrement 1. Nama variabelnya adalah "stress_level".

2. Membuat fungsi keanggotaan untuk setiap variabel:

```
# Membuat fungsi keanggotaan untuk setiap variabel fuzzy
snoring_rate['low'] = fuzz.trimf(snoring_rate.universe, [45, 45, 70.1])
snoring_rate['medium'] = fuzz.trimf(snoring_rate.universe, [45, 70.1, 100])
snoring_rate['high'] = fuzz.trimf(snoring_rate.universe, [70.1, 100, 100])

respiration_rate['low'] = fuzz.trimf(respiration_rate.universe, [16, 16, 21])
respiration_rate['medium'] = fuzz.trimf(respiration_rate.universe, [16, 21, 30])
respiration_rate['high'] = fuzz.trimf(respiration_rate.universe, [21, 30, 30])

body_temperature['low'] = fuzz.trimf(body_temperature.universe, [85, 85, 93])
body_temperature['medium'] = fuzz.trimf(body_temperature.universe, [85, 93, 99])
body_temperature['high'] = fuzz.trimf(body_temperature.universe, [93, 99, 99])

limb_movement['low'] = fuzz.trimf(limb_movement.universe, [4, 4, 11])
limb_movement['medium'] = fuzz.trimf(limb_movement.universe, [4, 11, 19])
limb_movement['high'] = fuzz.trimf(limb_movement.universe, [11, 19, 19])

blood_oxygen['low'] = fuzz.trimf(blood_oxygen.universe, [82, 82, 91])
blood_oxygen['medium'] = fuzz.trimf(blood_oxygen.universe, [82, 91, 97])
blood_oxygen['high'] = fuzz.trimf(blood_oxygen.universe, [91, 97, 97])

eye_movement['low'] = fuzz.trimf(eye_movement.universe, [60, 60, 90])
eye_movement['medium'] = fuzz.trimf(eye_movement.universe, [60, 90, 105])
eye_movement['high'] = fuzz.trimf(eye_movement.universe, [90, 105, 105])

sleeping_hours['low'] = fuzz.trimf(sleeping_hours.universe, [0, 0, 3.5])
sleeping_hours['medium'] = fuzz.trimf(sleeping_hours.universe, [0, 3.5, 9])
sleeping_hours['high'] = fuzz.trimf(sleeping_hours.universe, [3.5, 9, 9])

heart_rate['low'] = fuzz.trimf(heart_rate.universe, [50, 50, 62.5])
heart_rate['medium'] = fuzz.trimf(heart_rate.universe, [50, 62.5, 85])
heart_rate['high'] = fuzz.trimf(heart_rate.universe, [62.5, 85, 85])

stress_level['low'] = fuzz.trimf(stress_level.universe, [0, 1, 2])
stress_level['medium_low'] = fuzz.trimf(stress_level.universe, [2, 3, 4])
stress_level['medium'] = fuzz.trimf(stress_level.universe, [4, 5, 6])
stress_level['medium_high'] = fuzz.trimf(stress_level.universe, [6, 7, 8])
stress_level['high'] = fuzz.trimf(stress_level.universe, [8, 9, 10])
```

snoring_rate:

- Fungsi keanggotaan 'low': Rentang nilai 45 hingga 70.1, dengan puncak pada 45.
- Fungsi keanggotaan 'medium': Rentang nilai 45 hingga 100, dengan puncak pada 70.1.

- Fungsi keanggotaan 'high': Rentang nilai 70.1 hingga 100, dengan puncak pada 100.

respiration_rate:

- Fungsi keanggotaan 'low': Rentang nilai 16 hingga 21, dengan puncak pada 16.
- Fungsi keanggotaan 'medium': Rentang nilai 16 hingga 30, dengan puncak pada 21.
- Fungsi keanggotaan 'high': Rentang nilai 21 hingga 30, dengan puncak pada 30.

body_temperature:

- Fungsi keanggotaan 'low': Rentang nilai 85 hingga 93, dengan puncak pada 85.
- Fungsi keanggotaan 'medium': Rentang nilai 85 hingga 99, dengan puncak pada 93.
- Fungsi keanggotaan 'high': Rentang nilai 93 hingga 99, dengan puncak pada 99.

limb_movement:

- Fungsi keanggotaan 'low': Rentang nilai 4 hingga 11, dengan puncak pada 4.
- Fungsi keanggotaan 'medium': Rentang nilai 4 hingga 19, dengan puncak pada 11.
- Fungsi keanggotaan 'high': Rentang nilai 11 hingga 19, dengan puncak pada 19.

blood_oxygen:

- Fungsi keanggotaan 'low': Rentang nilai 82 hingga 91, dengan puncak pada 82.
- ungsi keanggotaan 'medium': Rentang nilai 82 hingga 97, dengan puncak pada 91.
- Fungsi keanggotaan 'high': Rentang nilai 91 hingga 97, dengan puncak pada 97.

eye_movement:

- Fungsi keanggotaan 'low': Rentang nilai 60 hingga 90, dengan puncak pada 60.
- Fungsi keanggotaan 'medium': Rentang nilai 60 hingga 105, dengan puncak pada 90.
- Fungsi keanggotaan 'high': Rentang nilai 90 hingga 105, dengan puncak pada 105.

sleeping_hours:

- Fungsi keanggotaan 'low': Rentang nilai 0 hingga 3.5, dengan puncak pada 0.
- Fungsi keanggotaan 'medium': Rentang nilai 0 hingga 9, dengan puncak pada 3.5.
- Fungsi keanggotaan 'high': Rentang nilai 3.5 hingga 9, dengan puncak pada 9.

heart_rate:

- Fungsi keanggotaan 'low': Rentang nilai 50 hingga 62.5, dengan puncak pada 50.
- Fungsi keanggotaan 'medium': Rentang nilai 50 hingga 85, dengan puncak pada 62.5.
- Fungsi keanggotaan 'high': Rentang nilai 62.5 hingga 85, dengan puncak pada 85.

stress_level:

- Fungsi keanggotaan 'low': Rentang nilai 0 hingga 2, dengan puncak pada 0.
- Fungsi keanggotaan 'medium_low': Rentang nilai 2 hingga 3, dengan puncak pada 4.
- Fungsi keanggotaan 'medium': Rentang nilai 4 hingga 5, dengan puncak pada 6.
- Fungsi keanggotaan 'medium_high': Rentang nilai 6 hingga 7, dengan puncak pada 8.
- Fungsi keanggotaan 'high': Rentang nilai 8 hingga 9, dengan puncak pada 10.

Pembahasan

1. Metode Mamdani :

```
# Membuat aturan fuzzy
# Definisikan aturan rule
rule1 = ctrl.Rule(snoring_rate['low'] & respiration_rate['low'] &
                  body_temperature['high'] & limb_movement['low'] &
                  blood_oxygen['high'] & eye_movement['low'] & sleeping_hours['high'] &
                  heart_rate['low'], stress_level['low'])
rule2 = ctrl.Rule(snoring_rate['low'] & respiration_rate['low'] &
                  body_temperature['high'] & limb_movement['medium'] &
                  blood_oxygen['high'] & eye_movement['medium'] & sleeping_hours['high'] &
                  heart_rate['low'], stress_level['medium_low'])
rule3 = ctrl.Rule(snoring_rate['medium'] & respiration_rate['medium'] &
                  body_temperature['medium'] & limb_movement['medium'] &
                  blood_oxygen['medium'] & eye_movement['medium'] & sleeping_hours['medium'] &
                  heart_rate['medium'], stress_level['medium'])
rule4 = ctrl.Rule(snoring_rate['high'] & respiration_rate['high'] &
                  body_temperature['medium'] & limb_movement['high'] &
                  blood_oxygen['low'] & eye_movement['high'] & sleeping_hours['low'] &
                  heart_rate['high'], stress_level['medium_high'])
rule5 = ctrl.Rule(snoring_rate['high'] & respiration_rate['high'] &
                  body_temperature['low'] & limb_movement['high'] &
                  blood_oxygen['low'] & eye_movement['high'] & sleeping_hours['low'] &
                  heart_rate['high'], stress_level['high'])
# Membuat sistem kontrol fuzzy
stress_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5])
stress_level_simulation = ctrl.ControlSystemSimulation(stress_ctrl)
```

Aturan fuzzy ini digunakan untuk menghubungkan kondisi-kondisi dari variabel fuzzy input dengan konklusi variabel fuzzy output. Berikut adalah penjelasan untuk setiap aturan fuzzy yang dibuat:

- Aturan 1: Jika snoring_rate rendah, respiration_rate rendah, body_temperature tinggi, limb_movement rendah, blood_oxygen tinggi, eye_movement rendah, sleeping_hours tinggi, dan heart_rate rendah, maka stress_level rendah.
- Aturan 2: Jika snoring_rate rendah, respiration_rate rendah, body_temperature tinggi, limb_movement sedang, blood_oxygen tinggi, eye_movement sedang, sleeping_hours tinggi, dan heart_rate rendah, maka stress_level medium rendah.
- Aturan 3: Jika snoring_rate sedang, respiration_rate sedang, body_temperature sedang, limb_movement sedang, blood_oxygen sedang, eye_movement sedang, sleeping_hours sedang, dan heart_rate sedang, maka stress_level medium.
- Aturan 4: Jika snoring_rate tinggi, respiration_rate tinggi, body_temperature sedang, limb_movement tinggi, blood_oxygen rendah, eye_movement tinggi, sleeping_hours rendah, dan heart_rate tinggi, maka stress_level medium tinggi.
- Aturan 5: Jika snoring_rate tinggi, respiration_rate tinggi, body_temperature rendah, limb_movement tinggi, blood_oxygen rendah, eye_movement tinggi, sleeping_hours rendah, dan heart_rate tinggi, maka stress_level tinggi.

Setelah aturan-aturan fuzzy didefinisikan, sistem kontrol fuzzy dibuat menggunakan objek ctrl.ControlSystem(). Aturan-aturan fuzzy tersebut kemudian disimpan dalam sistem kontrol fuzzy.

Selanjutnya, objek ctrl.ControlSystemSimulation() digunakan untuk membuat simulasi dari sistem kontrol fuzzy yang telah dibuat. Simulasi ini nantinya akan digunakan untuk melakukan inferensi fuzzy dan menghasilkan output berdasarkan kondisi input yang diberikan.

Dengan demikian, sistem kontrol fuzzy telah siap digunakan untuk melakukan evaluasi dan menghasilkan tingkat keanggotaan dari variabel fuzzy output (stress_level) berdasarkan kondisi-kondisi dari variabel fuzzy input.

```
def fuzzy_mamdani(snoring_rate_value, respiratory_rate_value, body_temperature_value,
                  limb_movement_value, blood_oxygen_value, eye_movement_value, sleeping_hours_value, heart_rate_value):
    # Mengatur input untuk sistem kontrol fuzzy dari dataset
    stress_level_simulation.input['snoring_rate'] = snoring_rate_value
    stress_level_simulation.input['respiration_rate'] = respiratory_rate_value
    stress_level_simulation.input['body_temperature'] = body_temperature_value
    stress_level_simulation.input['limb_movement'] = limb_movement_value
    stress_level_simulation.input['blood_oxygen'] = blood_oxygen_value
    stress_level_simulation.input['eye_movement'] = eye_movement_value
    stress_level_simulation.input['sleeping_hours'] = sleeping_hours_value
    stress_level_simulation.input['heart_rate'] = heart_rate_value

    # Melakukan perhitungan inferensi fuzzy
    stress_level_simulation.compute()

    # Mendapatkan hasil
    result = stress_level_simulation.output['stress_level']

    if result <= 2:
        category_level = 0
    elif result <= 4:
        category_level = 1
    elif result <= 6:
        category_level = 2
    elif result <= 8:
        category_level = 3
    else:
        category_level = 4

    return result, category_level
```

Berikut adalah penjelasan langkah-langkah yang dilakukan oleh fungsi fuzzy_mamdani:

1. Mengatur input untuk sistem kontrol fuzzy:

Nilai-nilai input dari variabel fuzzy (snoring_rate, respiratory_rate, body_temperature, limb_movement, blood_oxygen, eye_movement, sleeping_hours, dan heart_rate) diset dengan nilai yang diberikan sebagai argumen fungsi.

2. Melakukan perhitungan inferensi fuzzy:

Dengan menggunakan stress_level_simulation.compute(), sistem kontrol fuzzy melakukan inferensi fuzzy berdasarkan aturan-aturan fuzzy yang telah didefinisikan sebelumnya. Proses ini menghasilkan tingkat keanggotaan dari variabel fuzzy output (stress_level).

3. Mendapatkan hasil:

Hasil dari inferensi fuzzy, yaitu tingkat keanggotaan dari variabel fuzzy output (stress_level), disimpan dalam variabel result.

4. Mengkategorikan hasil:

Berdasarkan nilai result, dilakukan pemetaan ke dalam kategori stress_level dengan menggunakan serangkaian kondisi. Jika result kurang dari 2, maka kategori_level di-set sebagai 0. Jika result kurang dari 4, maka kategori_level di-set sebagai 1, dan seterusnya.

5. Mengembalikan hasil:

Fungsi mengembalikan nilai result (tingkat keanggotaan dari stress_level) dan category_level (kategori stress_level).

```
# Variabel untuk menghitung tingkat akurasi
total_data = len(data)
total_cocok = 0

# Iterasi melalui setiap baris data
for index, row in data.iterrows():
    snoring_rate_value = row['sr']
    respiratory_rate_value = row['rr']
    body_temperature_value = row['t']
    limb_movement_value = row['lm']
    blood_oxygen_value = row['bo']
    eye_movement_value = row['rem']
    sleeping_hours_value = row['sr2']
    heart_rate_value = row['hr']
    stress_level_value = row['sl']

    # Menggunakan fungsi fuzzy_mamdani untuk estimasi ukuran
    hasil, category_level = fuzzy_mamdani(snoring_rate_value, respiratory_rate_value, body_temperature_value,
                                          limb_movement_value, blood_oxygen_value, eye_movement_value,
                                          sleeping_hours_value, heart_rate_value)

    # Mencocokkan hasil yg diprediksi dengan dataset
    if stress_level_value == category_level:
        total_cocok += 1
    else:
        print("baris : " + str(index) + " hasil mamdani : " + str(category_level) +
              " stress level dataset : " + str(stress_level_value))
    # print(hasil)
# Hitung tingkat akurasi
print("banyak data : " + str(total_data))
print("data yang sama : " + str(total_cocok))
tingkat_akurasi = (total_cocok / total_data) * 100
print("Tingkat Akurasi: {:.2f}%".format(tingkat_akurasi))
```

6. Mempersiapkan variabel untuk menghitung tingkat akurasi:

Variabel total_data diinisialisasi dengan jumlah total baris dalam dataset, sedangkan total_cocok diinisialisasi dengan nilai awal 0. Variabel total_cocok akan digunakan untuk menghitung jumlah prediksi yang cocok dengan nilai sebenarnya.

7. Iterasi melalui setiap baris data:

Melalui perulangan for, kode tersebut mengiterasi melalui setiap baris dalam dataset. Untuk setiap baris, nilai-nilai input dari variabel fuzzy (snoring_rate, respiratory_rate, body_temperature, limb_movement, blood_oxygen, eye_movement, sleeping_hours, dan heart_rate) serta nilai target (stress_level) dari dataset diambil.

8. Menggunakan fungsi fuzzy_mamdani untuk estimasi ukuran:

Dengan menggunakan fungsi fuzzy_mamdani, nilai-nilai input dari variabel fuzzy dijadikan argumen untuk melakukan prediksi menggunakan sistem kontrol fuzzy yang telah dibuat sebelumnya. Fungsi ini mengembalikan hasil (tingkat keanggotaan dari stress_level) dan category_level (kategori stress_level) berdasarkan aturan-aturan fuzzy yang telah didefinisikan.

9. Mencocokkan hasil yang diprediksi dengan dataset:

Hasil prediksi category_level dibandingkan dengan nilai target stress_level dari dataset. Jika kedua nilai tersebut cocok, maka total_cocok akan ditambahkan satu.

10. Menghitung tingkat akurasi:

Setelah selesai mengiterasi melalui seluruh baris data, tingkat akurasi dihitung dengan membagi jumlah prediksi yang cocok (total_cocok) dengan total data (total_data) dan mengalikannya dengan 100. Hasilnya kemudian dicetak sebagai tingkat akurasi dalam bentuk persentase.

Output:

```
banyak data : 630
data yang sama : 535
Tingkat Akurasi: 84.920635%
```

11. Banyak Data:

Output ini mencetak jumlah total data yang ada dalam dataset. Ini memberikan informasi tentang ukuran dataset yang digunakan dalam evaluasi yaitu sebanyak 630 data.

12. Data yang Sama:

Output ini mencetak jumlah data yang hasil prediksinya cocok dengan nilai sebenarnya dalam dataset. Jumlah ini dihitung dengan menghitung berapa kali prediksi kategori stress_level yang dihasilkan oleh sistem kontrol fuzzy sama dengan nilai stress_level yang ada dalam dataset yaitu sebanyak 613 data yang sama.

13. Baris, Hasil Mamdani, Stress Level Dataset:

Output ini mencetak informasi per baris ketika prediksi kategori stress_level yang dihasilkan oleh sistem kontrol fuzzy tidak cocok dengan nilai stress_level yang ada dalam dataset. Ini membantu Anda untuk melihat baris data spesifik yang menghasilkan perbedaan tersebut dan membandingkan prediksi dengan nilai sebenarnya.

14. Tingkat Akurasi:

Output ini mencetak tingkat akurasi dari sistem kontrol fuzzy yang dihitung dengan membagi jumlah data yang hasil prediksinya cocok (total_cocok) dengan jumlah total data (total_data), kemudian dikalikan dengan 100 untuk menghasilkan nilai dalam bentuk persentase. Tingkat akurasi mengindikasikan sejauh mana prediksi sistem kontrol fuzzy cocok dengan nilai sebenarnya dalam dataset. Hasil dari akurasinya yaitu 84.920635%.

2. Metode Tsukamoto :

```
def fuzzy_tsukamoto(snoring_rate_value, respiratory_rate_value, body_temperature_value,
                    limb_movement_value, blood_oxygen_value, eye_movement_value, sleeping_hours_value, heart_rate_value):
    # Proses fuzzy
    snoring_rate_low_degree = fuzz.interp_membership(snoring_rate, snoring_rate_low, snoring_rate_value)
    snoring_rate_medium_degree = fuzz.interp_membership(snoring_rate, snoring_rate_medium, snoring_rate_value)
    snoring_rate_high_degree = fuzz.interp_membership(snoring_rate, snoring_rate_high, snoring_rate_value)

    respiration_rate_low_degree = fuzz.interp_membership(respiration_rate, respiration_rate_low, respiratory_rate_value)
    respiration_rate_medium_degree = fuzz.interp_membership(respiration_rate, respiration_rate_medium, respiratory_rate_value)
    respiration_rate_high_degree = fuzz.interp_membership(respiration_rate, respiration_rate_high, respiratory_rate_value)

    body_temperature_low_degree = fuzz.interp_membership(body_temperature, body_temperature_low, body_temperature_value)
    body_temperature_medium_degree = fuzz.interp_membership(body_temperature, body_temperature_medium, body_temperature_value)
    body_temperature_high_degree = fuzz.interp_membership(body_temperature, body_temperature_high, body_temperature_value)

    limb_movement_low_degree = fuzz.interp_membership(limb_movement, limb_movement_low, limb_movement_value)
    limb_movement_medium_degree = fuzz.interp_membership(limb_movement, limb_movement_medium, limb_movement_value)
    limb_movement_high_degree = fuzz.interp_membership(limb_movement, limb_movement_high, limb_movement_value)

    blood_oxygen_low_degree = fuzz.interp_membership(blood_oxygen, blood_oxygen_low, blood_oxygen_value)
    blood_oxygen_medium_degree = fuzz.interp_membership(blood_oxygen, blood_oxygen_medium, blood_oxygen_value)
    blood_oxygen_high_degree = fuzz.interp_membership(blood_oxygen, blood_oxygen_high, blood_oxygen_value)

    eye_movement_low_degree = fuzz.interp_membership(eye_movement, eye_movement_low, eye_movement_value)
    eye_movement_medium_degree = fuzz.interp_membership(eye_movement, eye_movement_medium, eye_movement_value)
    eye_movement_high_degree = fuzz.interp_membership(eye_movement, eye_movement_high, eye_movement_value)

    sleeping_hours_low_degree = fuzz.interp_membership(sleeping_hours, sleeping_hours_low, sleeping_hours_value )
    sleeping_hours_medium_degree = fuzz.interp_membership(sleeping_hours, sleeping_hours_medium, sleeping_hours_value )
    sleeping_hours_high_degree = fuzz.interp_membership(sleeping_hours, sleeping_hours_high, sleeping_hours_value )

    heart_rate_low_degree = fuzz.interp_membership(heart_rate, heart_rate_low, heart_rate_value )
    heart_rate_medium_degree = fuzz.interp_membership(heart_rate, heart_rate_medium, heart_rate_value )
    heart_rate_high_degree = fuzz.interp_membership(heart_rate, heart_rate_high, heart_rate_value )
```

Dalam fungsi fuzzy_tsukamoto, terdapat proses fuzzy menggunakan metode Tsukamoto. Metode ini melibatkan penggunaan fungsi keanggotaan untuk menghitung derajat keanggotaan setiap variabel fuzzy. Berikut adalah penjelasan langkah-langkah dalam proses tersebut:

1. Menghitung Derajat Keanggotaan Variabel Fuzzy:

Setiap variabel fuzzy memiliki beberapa himpunan linguistik dengan fungsi keanggotaan yang telah ditentukan sebelumnya, seperti 'low', 'medium', dan 'high'. Untuk setiap variabel fuzzy, derajat keanggotaan dihitung dengan menggunakan fungsi fuzz.interp_membership dengan argumen variabel fuzzy, himpunan linguistik, dan nilai variabel yang ingin dievaluasi. Misalnya, snoring_rate_low_degree akan menghitung derajat keanggotaan dari variabel snoring_rate pada himpunan linguistik 'low' dengan menggunakan nilai snoring_rate_value.

2. Langkah yang Sama untuk Setiap Variabel Fuzzy:

Proses yang sama berlaku untuk setiap variabel fuzzy yang terlibat dalam sistem kontrol fuzzy. Anda menghitung derajat keanggotaan untuk variabel respiration_rate, body_temperature, limb_movement, blood_oxygen, eye_movement, sleeping_hours, dan heart_rate dengan himpunan linguistik yang sesuai.

Proses ini membantu menentukan seberapa besar variabel input berkontribusi terhadap setiap himpunan linguistik dalam masing-masing variabel fuzzy. Derajat keanggotaan ini akan digunakan dalam tahap selanjutnya dari metode Tsukamoto untuk menghitung hasil output.

```
# Inferensi dengan menggunakan metode Tsukamoto
rule1 = np.fmin(snoring_rate_low_degree, np.fmin(respiration_rate_low_degree,
                np.fmin(body_temperature_high_degree, np.fmin(limb_movement_low_degree,
                np.fmin(blood_oxygen_high_degree, np.fmin(eye_movement_low_degree,
                np.fmin(sleeping_hours_high_degree, heart_rate_low_degree))))))
output1 = np.fmin(rule1, stress_level_low)

rule2 = np.fmin(snoring_rate_low_degree, np.fmin(respiration_rate_low_degree,
                np.fmin(body_temperature_high_degree, np.fmin(limb_movement_medium_degree,
                np.fmin(blood_oxygen_high_degree, np.fmin(eye_movement_medium_degree,
                np.fmin(sleeping_hours_high_degree, heart_rate_low_degree))))))
output2 = np.fmin(rule2, stress_level_medium_low)

rule3 = np.fmin(snoring_rate_medium_degree, np.fmin(respiration_rate_medium_degree,
                np.fmin(body_temperature_high_degree, np.fmin(limb_movement_medium_degree,
                np.fmin(blood_oxygen_medium_degree, np.fmin(eye_movement_medium_degree,
                np.fmin(sleeping_hours_medium_degree, heart_rate_medium_degree))))))
output3 = np.fmin(rule3, stress_level_medium)

rule4 = np.fmin(snoring_rate_high_degree, np.fmin(respiration_rate_high_degree,
                np.fmin(body_temperature_medium_degree, np.fmin(limb_movement_high_degree,
                np.fmin(blood_oxygen_low_degree, np.fmin(eye_movement_high_degree,
                np.fmin(sleeping_hours_low_degree, heart_rate_high_degree))))))
output4 = np.fmin(rule4, stress_level_medium_high)

rule5 = np.fmin(snoring_rate_high_degree, np.fmin(respiration_rate_high_degree,
                np.fmin(body_temperature_low_degree, np.fmin(limb_movement_high_degree,
                np.fmin(blood_oxygen_low_degree, np.fmin(eye_movement_high_degree,
                np.fmin(sleeping_hours_low_degree, heart_rate_high_degree))))))
output5 = np.fmin(rule5, stress_level_high)
```

3. Pengaplikasian Aturan Fuzzy:

Setiap aturan fuzzy didefinisikan menggunakan fungsi-fungsi minimum (np.fmin) untuk menghubungkan derajat keanggotaan variabel-input dan variabel-output dalam suatu aturan.

Misalnya, rule1 menggunakan fungsi-fungsi minimum untuk menghubungkan derajat keanggotaan dari variabel snoring_rate, respiration_rate, body_temperature, limb_movement, blood_oxygen, eye_movement, sleeping_hours, dan heart_rate dengan hipunan linguistik 'low' dari variabel-output stress_level_low.

```
# Penggabungan output dari semua aturan
aggregated = np.fmax(output1,
                     np.fmax(output2,
                         np.fmax(output3,
                             np.fmax(output4,output5))))

# Defuzzifikasi menggunakan metode Tsukamoto
stress_level_simulation = fuzz.defuzz(stress_level, aggregated, 'mom')

if stress_level_simulation <= 2:
    category_level = 0
elif stress_level_simulation <= 4:
    category_level = 1
elif stress_level_simulation <= 6:
    category_level = 2
elif stress_level_simulation <= 8:
    category_level = 3
else:
    category_level = 4

return stress_level_simulation, category_level
```

4. Penentuan Output Setiap Aturan:

Setiap aturan fuzzy menghasilkan variabel-output yang sesuai dengan tingkat kebenaran (derajat keanggotaan) aturan tersebut. Misalnya, output1 akan menyimpan tingkat kebenaran (derajat keanggotaan) aturan rule1 terhadap himpunan linguistik stress_level_low.

5. Penggabungan Output Aturan:

Output dari setiap aturan digabungkan menggunakan fungsi-fungsi maksimum (np.fmax) untuk mendapatkan hasil akhir inferensi.

Proses penggabungan dimulai dengan menggabungkan output1, output2, output3, output4, dan output5 menggunakan fungsi-fungsi maksimum berturut-turut.

Hasil akhir inferensi disimpan dalam variabel aggregated. Langkah-langkah ini menghasilkan output inferensi yang merupakan gabungan dari aturan-aturan fuzzy yang telah ditentukan sebelumnya.

6. Defuzzifikasi:

Hasil inferensi yang telah digabungkan (variabel aggregated) diolah menggunakan fungsi fuzz.defuzz() dengan metode Tsukamoto. Hasil defuzzifikasi disimpan dalam variabel stress_level_simulation.

7. Kategori Stress Level:

Hasil defuzzifikasi (stress_level_simulation) digunakan untuk menentukan kategori tingkat stres. Berdasarkan rentang nilai hasil defuzzifikasi, kategori tingkat stres ditentukan menggunakan serangkaian kondisi if-elif-else. Misalnya, jika stress_level_simulation kurang dari 2, kategori tingkat stres (category_level) diatur sebagai 0.

8. Output:

Hasil defuzzifikasi (stress_level_simulation) dan kategori tingkat stres (category_level) dikembalikan sebagai output dari fungsi. Langkah-langkah ini menghasilkan nilai stress level yang telah didefuzzifikasi menggunakan metode Tsukamoto serta kategori tingkat stres yang sesuai.



```
total_data = len(data)
total_cocok = 0
# Iterasi melalui setiap baris data
for index, row in data.iterrows():
    snoring_rate_value = row['sr']
    respiratory_rate_value = row['rr']
    body_temperature_value = row['t']
    limb_movement_value = row['lm']
    blood_oxygen_value = row['bo']
    eye_movement_value = row['rem']
    sleeping_hours_value = row['sr2']
    heart_rate_value = row['hr']
    stress_level_value = row['sl']

    # Menggunakan fungsi fuzzy_tsukamoto untuk estimasi ukuran
    hasil, category_level = fuzzy_tsukamoto(snoring_rate_value, respiratory_rate_value,
                                            body_temperature_value, limb_movement_value, blood_oxygen_value,
                                            eye_movement_value, sleeping_hours_value, heart_rate_value)

    if stress_level_value == category_level:
        total_cocok += 1
    else:
        print("baris : " + str(index) + " hasil tsukamoto : " + str(category_level) +
              " stress level dataset : " + str(stress_level_value))

    # print(hasil)

# Hitung tingkat akurasi
print("banyak data : " + str(total_data))
print("data yang sama : " + str(total_cocok))
tingkat_akurasi = (total_cocok / total_data) * 100
print("Tingkat Akurasi: {:.2f}%".format(tingkat_akurasi))
```

9. Mempersiapkan variabel untuk menghitung tingkat akurasi:

Variabel total_data diinisialisasi dengan jumlah total baris dalam dataset, sedangkan total_cocok diinisialisasi dengan nilai awal 0. Variabel total_cocok akan digunakan untuk menghitung jumlah prediksi yang cocok dengan nilai sebenarnya.

10. Iterasi melalui setiap baris data:

Melalui perulangan for, kode tersebut mengiterasi melalui setiap baris dalam dataset. Untuk setiap baris, nilai-nilai input dari variabel fuzzy (snoring_rate, respiratory_rate, body_temperature, limb_movement, blood_oxygen, eye_movement, sleeping_hours, dan heart_rate) serta nilai target (stress_level) dari dataset diambil.

11. Menggunakan fungsi fuzzy_mamdani untuk estimasi ukuran:

Dengan menggunakan fungsi fuzzy_mamdani, nilai-nilai input dari variabel fuzzy dijadikan argumen untuk melakukan prediksi menggunakan sistem kontrol fuzzy yang telah dibuat sebelumnya. Fungsi ini mengembalikan hasil (tingkat keanggotaan dari stress_level) dan category_level (kategori stress_level) berdasarkan aturan-aturan fuzzy yang telah didefinisikan.

12. Mencocokkan hasil yang diprediksi dengan dataset:

Hasil prediksi category_level dibandingkan dengan nilai target stress_level dari dataset. Jika kedua nilai tersebut cocok, maka total_cocok akan ditambahkan satu. Jika tidak cocok, maka baris data tersebut akan dicetak untuk menunjukkan hasil prediksi dan nilai stress_level sebenarnya.

13. Menghitung tingkat akurasi:

Setelah selesai mengiterasi melalui seluruh baris data, tingkat akurasi dihitung dengan membagi jumlah prediksi yang cocok (total_cocok) dengan total data (total_data) dan mengalikannya dengan 100. Hasilnya kemudian dicetak sebagai tingkat akurasi dalam bentuk persentase.

Output:

```
banyak data : 630
data yang sama : 524
Tingkat Akurasi: 83.174603%
```

14. Banyak Data:

Output ini mencetak jumlah total data yang ada dalam dataset. Ini memberikan informasi tentang ukuran dataset yang digunakan dalam evaluasi yaitu sebanyak 630 data.

15. Data yang Sama:

Output ini mencetak jumlah data yang hasil prediksinya cocok dengan nilai sebenarnya dalam dataset. Jumlah ini dihitung dengan menghitung berapa kali prediksi kategori stress_level yang dihasilkan oleh sistem kontrol fuzzy sama dengan nilai stress_level yang ada dalam dataset yaitu sebanyak 452 data yang sama.

16. Baris, Hasil Mamdani, Stress Level Dataset:

Output ini mencetak informasi per baris ketika prediksi kategori stress_level yang dihasilkan oleh sistem kontrol fuzzy tidak cocok dengan nilai stress_level yang ada dalam dataset. Ini membantu Anda untuk melihat baris data spesifik yang menghasilkan perbedaan tersebut dan membandingkan prediksi dengan nilai sebenarnya.

17. Tingkat Akurasi:

Output ini mencetak tingkat akurasi dari sistem kontrol fuzzy yang dihitung dengan membagi jumlah data yang hasil prediksinya cocok (total_cocok) dengan jumlah total data (total_data), kemudian dikalikan dengan 100 untuk menghasilkan nilai dalam bentuk persentase. Tingkat akurasi mengindikasikan sejauh mana prediksi sistem kontrol fuzzy cocok dengan nilai sebenarnya dalam dataset. Hasil dari akurasinya yaitu 83.174603%

KESIMPULAN

Berdasarkan hasil dan pembahan di atas, telah dilakukan studi kasus untuk menganalisis tingkat stres menggunakan metode fuzzy Mamdani dan metode fuzzy Tsukamoto. Data yang digunakan dalam studi kasus ini adalah data yang diambil dari keagle.

Hasil evaluasi menunjukkan bahwa metode fuzzy Mamdani memiliki tingkat akurasi sebesar 84.920635%. sementara metode fuzzy Tsukamoto memiliki tingkat akurasi sebesar 83.174603%

Dari hasil tersebut, dapat disimpulkan bahwa metode fuzzy Mamdani memberikan estimasi tingkat stres yang lebih akurat dibandingkan dengan metode fuzzy Tsukamoto dalam konteks studi kasus ini.

REFERENSI

[1] Hidayatullah, D. R, and Kusumadewi, S. “Rekomendasi tempat wisata menggunakan metode fuzzy logic.” Jurnal Teknik Pomits, vol. 4(1), no. 1-6, 2015, p. 16.

.

[2] Savitha, B. G, and Gopalakrishnan, E. A. “A comprehensive fuzzy-based restaurant recommendation system using data mining techniques.” International Journal of Advanced Science and Technology, vol. 28(8), 2019, pp. 219-228.

[3] Basori, W. I. (2018). Sistem Diagnosa Stres dan Rekam Medis Digital Berbasis Internet of Things Menggunakan Raspberry Pi. *Institut Teknologi Sepuluh Nopember, Surabaya*.

[4] Khairina, N. (2016). Analisis Fungsi Keanggotaan Fuzzy Tsukamoto Dalam Menentukan Status Kesehatan Tubuh Seseorang. *Sinkron: jurnal dan penelitian teknik informatika*, *1*(1), 19-19.

[5] S. Basriati, E. Safitri, and P. Nofridayani, “Penerapan Metode Fuzzy Tsukamoto dalam Menentukan Jumlah Produksi Tahu,” J. Sains Teknol. dan Ind., vol. 18, no. 1, pp. 120-125, 2020.

[6] Sihananto, A. N., Sari, A. P., Khariono, H., Fernanda, R. A., & Wijaya, D. C. M. (2022). Implementasi Metode K-Means Untuk Pengelompokan Kasus Covid-19 Tingkat Provinsi Di Indonesia. *Jurnal Informatika dan Sistem Informasi*, *3*(1), 76-85.

[7] Hamzah, M. A. (2016). Perancangan Alat Deteksi Tingkat Stress Dengan Metode Fuzzy Logic Berbasis Mikrokontroller Arduino. *JURNAL IT*, *7*(2), 123-132.

[8] Sugihartono, P. P. P., Hidayat, N., & Tibyani, T. (2020). Implementasi Metode Fuzzy Tsukamoto Untuk Deteksi Dini Tingkat Depresi Mahasiswa Yang Sedang Menempuh Skripsi (Studi Kasus:

Fakultas Ilmu Komputer Universitas Brawijaya). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *4*(10), 3432-3438.

[9] Rahakbauw, D. L., Afriananda, A., & Patty, H. W. (2022). Perbandingan Logika Fuzzy Metode Sugeno dan Metode Mamdani Untuk Deteksi Dini Penyakit Stroke. *Tensor: Pure and Applied Mathematics Journal*, *3*(1), 11-22.

[10] Plowerita, S., Hadi, I., Handayani, A. S., & Husni, N. L. (2021). Sistem Monitoring Kesehatan Dalam Penentuan Kondisi Tubuh Dengan Metode Fuzzy Mamdani. *PROtek: Jurnal Ilmiah Teknik Elektro*, *8*(2), 102-112.

[11] A. Wantoro, K. Muludi, and Sukisno, "Penerapan Logika Fuzzy pada Sistem Pendukung Keputusan Penentuan Kelayakan Kualitas Telur Bebek," JUTIS, vol. 7, no. 1, pp. 1-6, 2019, doi: 10.33592/jutis.Vol7.Iss1.140.

[12] Moi, M. A., Widodo, D., & Sutriningsih, A. (2017). HUBUNGAN GANGGUAN TIDUR DENGAN TEKANAN. *Nursing News*, 124-131.

[13] Kaban, F. A. B., Alamsyah, R., Sianturi, R., & Tambunan, K. (2022). Pemanfaatan metode fuzzy mamdani dalam deteksi penyakit manusia melalui gejala dan pola. *Jurnal Sains dan Teknologi Widyaloka (JSTekWid)*, *1*(1), 102-117.

[14] Fiano, D. S. I., & Purnomo, A. S. (2017). Sistem Pakar Untuk Mendeteksi Tingkat Resiko Penyakit Jantung Dengan Fuzzy Inferensi (Mamdani). *INFORMAL: Informatics Journal*, *2*(2), 64-78.

[15] Nawindah, S. L., & Lydiani, S. (2021). FUZZY TSUKAMOTO UNTUK DETEKSI TINGKAT KESEHATAN MENTAL MAHASISWA. *Jurnal Ilmiah Teknologi Informasi Terapan*, *7*(2), 159.

[16] Hidayati, L. N., & Harsono, M. (2021). TINJAUAN LITERATUR MENGENAI STRES DALAM ORGANISASI. *Jurnal Ilmu Manajemen, Volume 18, Nomor 1, 2021, 18*, 20-30.