

BAB II

KAJIAN LITERATUR

2.1 Penelitian Terdahulu

Penelitian terdahulu berfungsi sebagai tolak ukur teknis sekaligus sebagai landasan empiris untuk mengidentifikasi celah penelitian. Pada penelitian ini, pemetaan penelitian terdahulu difokuskan pada dua domain utama: efektivitas aplikasi manajemen keuangan personal dan implementasi kecerdasan buatan berbasis *Fuzzy Logic*.

Penelitian pertama dilakukan oleh Kristina dkk. [6] yang membangun aplikasi manajemen keuangan pribadi bernama *BudgetApp*. Sistem tersebut dikembangkan menggunakan bahasa pemrograman *Java* dengan fokus utama pada fungsionalitas pencatatan transaksi dan pelaporan visual bulanan. Meskipun *BudgetApp* terbukti secara empiris mampu menumbuhkan kebiasaan positif pengguna dalam melacak arus kas masuk dan keluar, penelitian tersebut menyimpulkan adanya kelemahan fatal pada perilaku pengguna. Pengguna masih sangat sering melakukan pengeluaran berlebih. Hal ini terjadi karena ketiadaan fitur rekomendasi cerdas; sistem hanya melaporkan defisit setelah uang tersebut habis, alih-alih memberikan panduan preventif atau pedoman alokasi anggaran sebelum pengguna membelanjakan uangnya.

Di ranah kecerdasan buatan, penelitian kedua oleh Alhafiz dan Sriani [7] mengeksplorasi secara mendalam penggunaan metode *Fuzzy Tsukamoto* dalam sebuah Sistem Pendukung Keputusan (SPK). Penelitian tersebut membuktikan bahwa algoritma *Tsukamoto* memiliki tingkat akurasi komputasi dan presisi yang sangat tinggi saat digunakan untuk memetakan profil kompetensi akademik menjadi rekomendasi mata kuliah pilihan bagi mahasiswa. Keandalan metode ini dibuktikan dengan tingkat persentase kesalahan (*Mean Absolute Percentage Error / MAPE*) yang sangat rendah pada pengujian sistem, yakni hanya sebesar 0,1096%. Logika *fuzzy* terbukti sangat adaptif dan objektif dalam menangani kondisi penilaian atau kriteria masukan yang tidak mutlak (hitam-putih), seperti memetakan kemampuan mahasiswa ke dalam batasan parameter yang samar seperti "lemah", "lumayan", dan "kuat".

Sejalan dengan eksplorasi algoritma tersebut, Baraputri dkk. [8] juga menerapkan metode *Fuzzy Tsukamoto*, kali ini ke dalam sebuah kerangka sistem pakar untuk diagnosis medis. Penelitian ini secara spesifik menyoroti keunggulan arsitektur penalaran metode *Tsukamoto* yang dinilai memiliki keunggulan matematis dibandingkan metode *Mamdani*

atau *Sugeno*. Karena *Tsukamoto* menghasilkan keluaran berbasis persamaan monoton yang lebih linier dan terstruktur, metode ini dinyatakan sangat kompatibel dan efisien dari segi beban komputasi memori saat diimplementasikan ke dalam kerangka kerja (*framework*) *back-end* asinkron seperti *Node.js*.

Berdasarkan sintesis dari tinjauan literatur di atas, terlihat adanya *research gap* yang saling bersinggungan. Aplikasi keuangan yang ada saat ini umumnya terjebak pada fungsi administratif sebagai alat pencatat statis yang reaktif [6]. Di sisi lain, dari perspektif rekayasa perangkat lunak, pengembangan sistem cerdas seringkali dihadapkan pada tantangan kesenjangan antara rancangan arsitektur perangkat lunak dengan implementasi aktual pada kode sumbernya [9]. Menurut penelitian Tian dkk. [9], ketidakselarasan antara keputusan desain tingkat tinggi dan penulisan kode tingkat rendah ini berpotensi besar menghambat proses pemeliharaan serta evolusi sistem di masa depan. Oleh karena itu, pengembangan aplikasi *Smart Pocket* dirancang tidak hanya untuk mengintegrasikan algoritma *Fuzzy Tsukamoto* secara cerdas pada masalah finansial, melainkan juga dibangun dengan memastikan tata arsitektur sistem berjalan selaras dengan penulisan kode sumbernya, sehingga stabilitas komputasi dan skalabilitas aplikasi dapat terjamin dengan baik [8].

Oleh karena itu, kebaruan dari penelitian ini terletak pada penggabungan inovatif dari kedua ranah tersebut, yakni kecerdasan buatan dan rekayasa perangkat lunak. Penelitian ini mengusulkan penerapan ketajaman dan fleksibilitas algoritma *Fuzzy Tsukamoto* sebagai mesin inferensi ke dalam aplikasi manajemen keuangan harian, yang dibangun di atas fondasi arsitektur sistem dan kode sumber yang terintegrasi secara selaras. Tujuannya adalah untuk mentransformasi aplikasi pencatat yang tadinya pasif menjadi sistem rekomendasi proaktif. Keunggulan implementasi metode *Tsukamoto* dalam sebuah sistem pendukung keputusan ini juga diperkuat secara empiris oleh studi komparatif yang membandingkan performa algoritma *Mamdani*, *Sugeno*, dan *Tsukamoto* [10]. Penelitian tersebut menyimpulkan bahwa metode *Tsukamoto* memiliki persentase tingkat kesalahan (*Mean Absolute Percentage Error/MAPE*) yang paling kecil dan tingkat kebenaran tertinggi dibandingkan dua metode lainnya [10]. Selain itu, metode *Tsukamoto* sangat ideal diterapkan pada sistem rekomendasi karena memberikan kejelasan nilai keluaran yang bersifat tegas (*crisp*) dan transparansi pada proses pengambilan keputusannya [7].

2.2 Pengembangan Aplikasi Berbasis Web

Dalam aturan ilmu Teknik Informatika, pengembangan aplikasi merupakan wujud pelaksanaan langsung dari rekayasa perangkat lunak (*Software Engineering*). Penulisan kode

program (*coding*) adalah salah satu cara untuk membangun aplikasi, tetapi tidak hanya itu, melainkan mencakup siklus sistematis yang melibatkan beberapa perancangan arsitektur sistem, pemilihan struktur data yang efisien, optimasi algoritma, hingga pemeliharaan sistem purna-rilis untuk membangun perangkat lunak yang andal, aman, dan dapat memberikan solusi komputasi atas suatu permasalahan dunia nyata [11].

Aplikasi berbasis web adalah perangkat lunak terdistribusi yang diakses melalui peramban web dengan memanfaatkan jaringan internet atau intranet. Arsitektur aplikasi web modern umumnya beroperasi pada model *Client-Server* yang memisahkan tanggung jawab antara antarmuka pengguna dan logika pemrosesan data.

1. Sisi Klien: Peramban web bertindak sebagai klien yang merender antarmuka pengguna (UI) menggunakan kombinasi *HyperText Markup Language* (HTML) untuk struktur, *Cascading Style Sheets* (CSS) untuk tata letak visual, dan *JavaScript* untuk interaktivitas dinamis. Klien bertugas menangkap *input* pengguna dan mengirimkan permintaan protokol HTTP/HTTPS ke *server*.
2. Sisi *Server*: Di belakang layar, bahasa pemrograman berkinerja tinggi bertugas menerima permintaan klien, memproses logika bisnis yang kompleks dan berinteraksi dengan sistem manajemen basis data menggunakan kueri spesifik (seperti SQL). Setelah data diproses, *server* akan mengembalikan respons, umumnya dalam format *JavaScript Object Notation* (JSON), kembali ke klien [12].

Keunggulan utama dari pengembangan aplikasi berbasis web adalah sifatnya yang *platform-independent* dan *cross-compatible*. Hal ini menandakan bahwa pengguna tidak perlu mengunduh *file installer* atau melakukan instalasi perangkat lunak secara lokal di *hard drive* mereka. Selama perangkat memiliki peramban dan konektivitas jaringan, aplikasi dapat diakses secara *real-time*. Model ini juga sangat memudahkan proses *deployment* dan *maintenance* bagi pengembang, karena setiap pembaruan atau perbaikan *bug* yang dilakukan di sisi *server* akan secara otomatis dan instan dirasakan oleh seluruh pengguna tanpa mengharuskan mereka melakukan pembaruan aplikasi secara manual [13].

2.3 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak adalah sebuah *framework* terstruktur yang digunakan untuk menyusun, merencanakan, dan mengendalikan tahapan proses pembuatan sistem informasi atau aplikasi. Pada penelitian ini, pengembangan aplikasi *Smart Pocket* menggunakan kerangka kerja *Software Development Life Cycle* (SDLC) dengan mengadopsi model *Incremental*.

Model *Incremental* adalah metode pengembangan adaptif di mana produk dirancang, diimplementasikan, dan diuji secara bertahap hingga produk tersebut mencapai fungsionalitas penuh [14]. Secara metodologis, setiap siklus pengembangan aplikasi *Smart Pocket* akan melewati empat tahapan rekayasa perangkat lunak dasar sebagai berikut:

1. Analisis Sistem

Tahap ini merupakan fase identifikasi dan pemetaan kebutuhan sistem yang dirangkum berdasarkan kondisi nyata. Analisis ini mencakup:

- Analisis Sistem Berjalan: Mengevaluasi proses manajemen keuangan pengguna yang saat ini masih dilakukan secara manual dan konvensional tanpa panduan pakar.
- Analisis Masalah: Mengidentifikasi kelemahan sistem berjalan, yaitu kekakuan paradigma anggaran persentase tetap (*rigid budgeting*) dan tingginya kerentanan kesalahan pencatatan oleh manusia (*human error*).
- Analisis Kebutuhan: Merumuskan spesifikasi sistem secara terperinci, baik kebutuhan fungsional (seperti fitur autentikasi, manajemen kas, dan integrasi algoritma *Fuzzy Tsukamoto*), maupun kebutuhan non-fungsional (seperti keamanan enkripsi Bcrypt dan performa berbasis web).

2. Perancangan Sistem

Tahap ini bertujuan untuk mentransformasikan dokumen analisis menjadi cetak biru (*blueprint*) teknis yang siap diimplementasikan oleh pemrogram. Fokus perancangan pada pengembangan ini dibatasi pada dua elemen arsitektural utama, yaitu:

- Perancangan Antarmuka (*User Interface*): Pembuatan desain tata letak visual (*mock-up*) antarmuka aplikasi yang intuitif bagi pengguna.
- Perancangan Basis Data: Pemodelan struktur penyimpanan data relasional menggunakan *Entity Relationship Diagram* (ERD) beserta Kamus Data (*Data Dictionary*) untuk menjamin integritas rekam transaksi finansial.

3. Konstruksi dan Pengkodean

Tahap ini adalah proses menerjemahkan rancangan teknis ke dalam baris-baris instruksi bahasa pemrograman. Ekosistem sisi klien (*Front-End*) dibangun menggunakan HTML5, CSS3, dan *JavaScript*, sedangkan sisi server (*Back-end*) dibangun menggunakan runtime environment *Node.js* untuk memproses logika *Fuzzy* dan berinteraksi dengan basis data *MySQL*.

4. Pengujian

Tahap ini merupakan fase validasi untuk memastikan seluruh fitur berjalan sesuai spesifikasi awal. Pendekatan pengujian yang digunakan dalam penelitian ini adalah metode *Black Box Testing* dengan teknik *Equivalence Partitioning* dan *Boundary Value Analysis*. Pengujian difokuskan pada pengujian fungsionalitas antarmuka (user acceptance) dan validasi akurasi perhitungan matematis mesin inferensi *Fuzzy Tsukamoto* guna memastikan sistem bebas dari bug atau eror komputasi (seperti pembagian angka nol atau *infinite loop*).

2.4 Sistem Pakar (*Expert System*)

Sistem pakar merupakan salah satu cabang dari disiplin keilmuan kecerdasan buatan yang secara khusus dirancang untuk merepresentasikan dan meniru kemampuan analisis serta pengambilan keputusan layaknya seorang pakar atau spesialis manusia [15]. Tujuan utama dari sistem ini adalah untuk mentransfer keahlian langka dan mahal dari seorang profesional manusia ke dalam wujud komputasional yang dapat diakses oleh masyarakat luas kapan saja.

Arsitektur sistem pakar pada umumnya terdiri dari beberapa komponen esensial, yang paling utama adalah:

1. Basis Pengetahuan (*Knowledge Base*): Repositori yang berisi sekumpulan aturan heuristik, fakta, dan pemahaman keahlian yang diekstraksi dari pakar manusia.
2. Mesin Inferensi (*Inference Engine*): Komponen otak pemrosesan yang bertugas menavigasi basis pengetahuan, mengevaluasi kondisi *input* pengguna menggunakan pendekatan deduktif atau induktif, untuk menghasilkan kesimpulan logis.
3. Memori Kerja (*Working Memory*): Ruang penyimpanan sementara untuk fakta-fakta spesifik dari pengguna yang sedang dievaluasi oleh sistem.

Dalam konteks arsitektur aplikasi *Smart Pocket*, sistem pakar bertindak secara virtual untuk menggantikan peran seorang konsultan perencana keuangan. Alih-alih membayar biaya konsultasi, pengguna dapat memasukkan kondisi finansialnya, dan mesin inferensi akan mencocokkannya dengan aturan-aturan perencanaan keuangan yang sehat.

2.5 Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan (SPK) adalah sebuah sistem informasi berbasis komputer yang sangat interaktif, dirancang secara khusus untuk membantu pembuat keputusan memanfaatkan kombinasi data, dokumen, pengetahuan, serta model komputasi

matematis dalam memecahkan masalah yang bersifat semi-terstruktur atau tidak terstruktur sama sekali [16].

Berbeda dengan Sistem Pemrosesan Transaksi (TPS) yang hanya bertugas merekam aliran data operasional secara mutlak, SPK tidak bertujuan untuk mengambil alih kendali keputusan secara seutuhnya dari tangan manusia. SPK menyadari bahwa pengambilan keputusan manusia sering kali melibatkan faktor emosional dan intuisi. Oleh karena itu, SPK hadir untuk memperluas kapabilitas kognitif manusia dengan memberikan panduan, prediksi, atau rekomendasi yang didukung oleh perhitungan matematis yang bebas dari bias subjektivitas. Masalah penganggaran bulanan adalah contoh sempurna dari masalah semi-terstruktur, di mana data kuantitatif (jumlah gaji) bercampur dengan variabel kualitatif (tuntutan gaya hidup atau tanggung jawab keluarga).

2.6 Sistem Rekomendasi

Sistem rekomendasi merupakan bagian integral dari kerangka sistem penyaringan informasi yang berevolusi dengan tujuan untuk memprediksi tingkat preferensi, utilitas, atau kelayakan suatu entitas bagi pengguna tertentu di tengah luapan informasi [17]. Secara konvensional, sistem rekomendasi dibagi menjadi dua jenis utama, yaitu *Collaborative Filtering* (berdasarkan kemiripan antar-pengguna) dan *Content-Based Filtering* (berdasarkan atribut item).

Namun, pada penelitian ini, sistem yang dibangun mengadopsi pendekatan *Knowledge-Based Recommendation System* (Sistem Rekomendasi Berbasis Pengetahuan). Pendekatan ini tidak memerlukan data riwayat pengguna lain atau rating untuk bisa bekerja, melainkan mengandalkan pengetahuan domain yang spesifik. Sistem difungsikan untuk menganalisis profil finansial demografis pengguna dan langsung menyajikan saran distribusi persentase dana secara tripartit: Kebutuhan (keperluan absolut), Keinginan (fleksibilitas konsumtif), dan Tabungan (ketahanan masa depan). Rekomendasi ini dirancang agar bertindak sebagai batas psikologis pelindung yang paling rasional untuk mencegah defisit anggaran bulanan.

2.7 Algoritma Fuzzy Logic

2.7.1 Konsep Dasar Logika Fuzzy

Logika Fuzzy pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965 sebagai sebuah metodologi komputasi yang meniru cara berpikir dan penalaran manusia. Berbeda dengan logika tegas (*crisp/Boolean logic*) konvensional komputer yang hanya

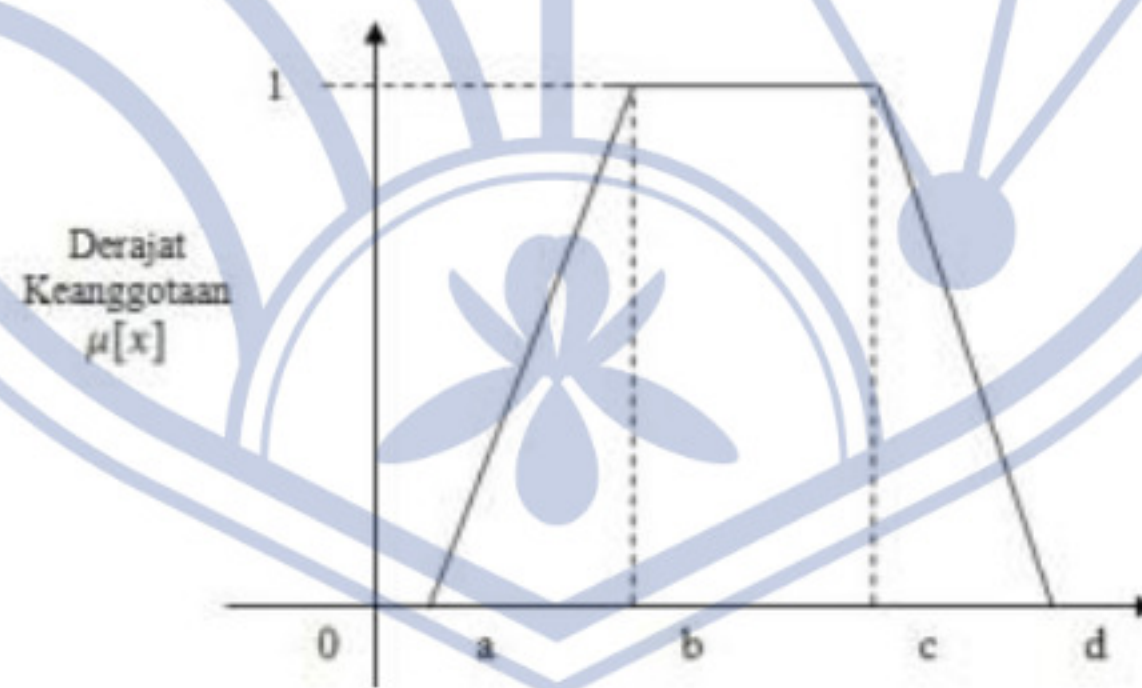
mengenai dua nilai mutlak yaitu “Benar” (1) atau “Salah” (0), logika *fuzzy* memiliki rentang nilai kebenaran yang kontinu antara 0 hingga 1 [18].

Logika ini secara luas diadopsi oleh para ilmuwan, peneliti, dan pengembang perangkat lunak kecerdasan buatan di seluruh dunia. Logika *fuzzy* sangat diperlukan ketika sebuah sistem komputer harus menghadapi masalah yang ambigu, samar, dan mengandung unsur ketidakpastian. Dalam bahasa manusia, ukuran seperti “gaji yang cukup”, “utang yang besar”, atau “suhu yang panas” memiliki makna yang relatif bagi tiap orang dan tidak bisa dipotong secara tegas dengan satu angka pasti.

Cara kerja logika *fuzzy* adalah dengan memetakan variabel linguistik manusia tersebut ke dalam suatu besaran matematis. Melalui derajat keanggotaan, sebuah nilai *input* tidak langsung dikategorikan murni ke dalam satu kelompok saja, melainkan bisa memiliki sebagian sifat dari kelompok A dan sebagian sifat dari kelompok B secara bersamaan, sehingga menghasilkan transisi keputusan yang sangat halus dan adaptif [18].

2.7.2 Fungsi Keanggotaan (*Membership Function*)

Fungsi keanggotaan adalah sebuah kurva grafis matematis yang memetakan titik-titik nilai *input* data tegas ke dalam derajat keanggotaan (dilambangkan dengan μ) yang bernilai pada interval $[0, 1]$. Terdapat berbagai jenis bentuk kurva yang umum digunakan, seperti kurva segitiga, kurva trapesium, dan kurva linier.



Gambar 1.1 Grafik Fungsi Keanggotaan

Dalam konteks perancangan aplikasi "*Smart Pocket*", penerapan fungsi keanggotaan khususnya pemanfaatan kurva representasi linier naik dan linier turun memiliki urgensi yang amat sangat krusial. Konsep penilaian dalam ilmu keuangan sangatlah subjektif, ambigu, dan relatif antarindividu.

Sebagai contoh nyata, uang tunai dengan nominal Rp4.000.000,00 mungkin akan dikategorikan secara absolut sebagai "Pendapatan Tinggi" bagi seorang mahasiswa lajang yang tinggal di kota kecil, tetapi akan dengan cepat dianggap sebagai "Pendapatan Rendah" bagi seorang kepala keluarga dengan tiga anak usia sekolah yang berdomisili di wilayah metropolitan.

Melalui fungsi keanggotaan linier, sistem komputasi mampu menghilangkan kesalahpahaman subjektif tersebut. Nilai tegas berupa angka nominal riil (Rupiah) dikonversi secara matematis menjadi derajat kelayakan atau intensitas (*fuzzification*). Dengan demikian, mesin komputer dapat mengukur dan menetapkan seberapa "Tinggi" atau seberapa "Rendah" kondisi finansial pengguna tersebut secara proporsional dan bertahap, menghindari lompatan keputusan yang terlalu drastis apabila terjadi perubahan kecil pada gaji pengguna.

2.7.3 Logika Fuzzy Metode Tsukamoto

Diperkenalkan oleh Dr. Y. Tsukamoto, metode Tsukamoto merupakan teknik penalaran fuzzy terstruktur di mana setiap konsekuensi harus direpresentasikan menggunakan himpunan fuzzy dengan fungsi keanggotaan yang monoton (hanya linier naik atau linier turun, tanpa ada elemen segitiga di bagian keluaran). Karakteristik ini membuat perhitungan defuzzifikasinya lebih efisien. Penyelesaian komputasi pada metode Tsukamoto selalu melewati empat tahapan matematis sistematis [8], yaitu:

1. Fuzzifikasi. Merupakan gerbang pertama komputasi. Proses ini melibatkan perubahan variabel tegas (*crisp input*) berupa angka riil dari pengguna (seperti Rp 5.000.000) menjadi variabel fuzzy (derajat keanggotaan linguistik, seperti $\mu_{Rendah} = 0,4$ dan $\mu_{Tinggi} = 0,6$) berdasarkan himpunan dan domain kurva yang telah ditentukan dan dikonfigurasi sebelumnya pada sistem.
2. Pembentukan Basis Aturan (*Rule Base*). Ini adalah inti dari ekstraksi kepakaran. Aturan-aturan ini disusun menggunakan logika proposisional dengan format *IF THEN*. Pada desain penelitian ini, terdapat 5 variabel masukan independen (Pendapatan, Pengeluaran, Cicilan, Tanggungan, dan Status). Berdasarkan probabilitas kombinatorial, integrasi himpunan dari kelima variabel tersebut akan menghasilkan sebuah matriks ruang keputusan yang terdiri dari 32 basis aturan ($2 \times 2 \times 2 \times 2 \times 2 = 32$) guna menampung secara komprehensif seluruh skenario kondisi kehidupan pengguna.

3. Mesin Inferensi dan Pencarian Nilai *Alpha* (α). Pada tahap komputasi inferensi ini, sistem mengevaluasi seluruh 32 aturan yang ada. Metode ini secara ketat menggunakan operator logika konjungsi (*AND*) dalam menggabungkan premis. Oleh karenanya, untuk mencari nilai kekuatan atau predikat dari setiap aturan (*Alpha-predikat*, dilambangkan dengan α), sistem harus mencari nilai interseksi terkecil dari himpunan-himpunan tersebut. Rumus perhitungannya menggunakan fungsi minimum (*MIN*) dari semua anteseden yang menyala di dalam aturan:

$$\alpha_{predikat_i} = \min(\mu_{x_1}, \mu_{x_2}, \mu_{x_3}, \dots, \mu_{x_n})$$

Keterangan:

$\alpha_{predikat_i}$: Nilai kekuatan, bobot pemicu (*firing strength*) dari aturan ke-*i*.

μ_x : Derajat keanggotaan individual dari masing-masing variabel masukan.

Setelah nilai predikat yang merupakan wakil terkecil didapatkan, nilai α tersebut diimplikasikan ke dalam fungsi keluaran kurva monoton untuk mencari besaran nilai konsekuen parsial (z_i) dari aturan tersebut (seperti berapa persen Kebutuhan menurut Aturan 1).

4. Defuzzifikasi. Merupakan tahap penyatuan akhir untuk mengembalikan nilai-nilai *fuzzy* majemuk menjadi satu nilai tegas tunggal (*crisp output*) yang dapat dipahami dan diaplikasikan langsung oleh pengguna. Karena keluaran *Tsukamoto* bersifat monoton, perhitungan akhirnya menggunakan metode Rata-Rata Terpusat (*Center of Average / Weighted Average*). Metode ini jauh lebih merepresentasikan titik keseimbangan sistem dibandingkan metode nilai rata-rata maksimum (*Mean of Maximum*), dengan rumusan matematis:

$$Z = \frac{\sum_{i=1}^n (\alpha_i \cdot z_i)}{\sum_{i=1}^n \alpha_i}$$

Keterangan:

Z : Hasil keluaran tegas akhir (Persentase Alokasi Dana).

α_i : Nilai *alpha-predikat* dari aturan aktif ke-*i*.

z_i : Nilai implikasi konsekuen dari aturan aktif ke-*i*

n : Total jumlah aturan (*rule*) dalam matriks yang dievaluasi.

2.8 Pengelolaan Keuangan Pribadi

2.8.1 Konsep Literasi dan Inklusi Keuangan

Pengelolaan keuangan pribadi merupakan ilmu sekaligus seni dalam merencanakan, mengelola, dan mengendalikan sumber daya uang yang dimiliki oleh individu atau keluarga. Fondasi utama dari pengelolaan ini bertumpu pada dua hal, yaitu literasi keuangan (tingkat pemahaman seseorang terhadap konsep risiko dan pengelolaan dana) serta inklusi keuangan (tingkat ketersediaan akses masyarakat terhadap layanan lembaga keuangan) [1].

Survei terbaru dari Otoritas Jasa Keuangan (OJK), menunjukkan masih ada kesenjangan. Inklusi keuangan masyarakat Indonesia memang lebih tinggi, tetapi literasi keuangan masih rendah. Jadi hal ini memicu permasalahan krisis finansial bagi banyak orang, terutama ketika beban biaya hidup terus melonjak seiring dengan inflasi tahunan. Banyak orang bisa mendapat pinjaman atau kredit, namun mereka tidak mengatur arus kasnya sehingga terjatuh dalam siklus utang yang buruk [1].

Untuk mengatasi permasalahan tersebut, pengelolaan arus kas harus dilakukan secara disiplin dan didukung oleh alat bantu yang cerdas. Individu dituntut untuk secara konsisten mencatat seluruh sumber pendapatan bulanan, melacak setiap titik pengeluaran, menghitung beban cicilan, serta menyisihkan dana darurat. Penggunaan sistem pakar diharapkan mampu mengotomatiskan dan mempermudah proses evaluasi tersebut [2].

2.8.2 Metode Penganggaran (*Budgeting*) Konvensional

Metode penganggaran adalah kerangka kerja sistematis yang digunakan untuk memetakan ke mana arah uang akan dihabiskan atau disimpan. Salah satu metode yang sangat populer dan sering diajarkan oleh para perencana keuangan adalah metode 50/30/20. Metode ini membagi total pendapatan ke dalam tiga pos utama, yakni 50% untuk kebutuhan pokok (*needs*), 30% untuk keinginan atau hiburan (*wants*), dan 20% sisanya untuk tabungan atau investasi (*savings*) [19].

Metode ini umumnya diterapkan oleh para pekerja, pegawai, atau keluarga muda dalam mengatur gaji bulanan mereka. Metode 50/30/20 idealnya dipraktikkan pada awal bulan sesaat setelah menerima penghasilan, untuk mencegah uang habis tanpa jejak sebelum akhir bulan. Tujuannya adalah untuk menciptakan batas psikologis agar seseorang tidak menghabiskan uang melebihi porsinya.

Namun, cara penerapan metode konvensional ini bersifat sangat kaku dan linier. Metode ini memaksa pembagian persentase mutlak tanpa mempertimbangkan variabel

dinamis kehidupan manusia, seperti jumlah anggota keluarga yang menjadi tanggungan atau beban utang masa lalu yang harus dilunasi [19]. Oleh karena itu, diperlukan sebuah sistem rekomendasi cerdas yang dapat menggeser persentase tersebut secara proporsional sesuai dengan kondisi nyata individu.

