

BAB IV

PERANCANGAN

4.1 Analisis Sistem terdahulu

Mekanisme yang telah ada sebelumnya menunjukkan bahwa pendekatan dalam pemilihan laptop umumnya masih menggunakan metode konvensional yang kurang mampu menangani ketidakpastian dan subjektivitas dalam pengambilan keputusan. Sistem terdahulu memiliki beberapa karakteristik dan keterbatasan yang perlu dipahami untuk mengembangkan sistem yang lebih baik.

4.2 Sistem Konvensional

Sistem pemilihan laptop tradisional biasanya menerapkan metode yang berdasarkan kriteria dengan penilaian yang dasar. Pendekatan ini menerapkan teknik serupa Simple Additive Weighting (SAW), Analytical Hierarchy Process (AHP), atau TOPSIS. Walaupun cara-cara ini relatif efektif, mereka memiliki batasan dalam mengatasi ketidakpastian dan nilai-nilai bahasa yang sering muncul dalam penilaian konsumen.

Perbandingan Konsep AHP dan SAW:

- a) Memerlukan perhitungan yang lebih kompleks (AHP) sedangkan Perhitungan yang sederhana dan mudah dipahami (SAW).
- b) Dalam bentuk Fuzzy AHP (FAHP), dapat menangani ketidakpastian dengan baik namun Dalam bentuk Fuzzy SAW (FSAW), memerlukan konversi bilangan fuzzy ke crisp.
- c) Jika AHP mempunyai mekanisme consistency ratio (CR) untuk memvalidasi penilaian sedangkan SAW Tidak memiliki mekanisme validasi konsistensi built-in.

Kelebihan Sistem Konvensional:

- a) Perhitungan yang relatif sederhana dan mudah dipahami;
- b) Implementasi yang *straight forward* dengan kompleksitas komputasi rendah;
- c) Hasil yang *deterministic* dan dapat diprediksi.

Keterbatasan Sistem Konvensional:

- a) Tidak dapat menangani ketidakpastian dan ambiguitas dalam penilaian;
- b) Sulit mengakomodasi preferensi pengguna yang bersifat linguistik;
- c) Sensitif terhadap perubahan bobot kriteria;

Kurang fleksibel dalam menangani kriteria yang bersifat kualitatif.

4.2.1 Sistem Fuzzy Terdahulu

Beberapa penelitian sebelumnya telah mengimplementasikan logika fuzzy untuk pemilihan laptop dengan berbagai pendekatan. Analisis terhadap sistem-sistem ini menunjukkan bahwa meskipun mampu menangani ketidakpastian, masih terdapat ruang untuk perbaikan dalam hal akurasi, efisiensi, dan kemudahan penggunaan.

Taufik et al. (2018) mengembangkan sistem dengan kombinasi Fuzzy Mamdani dan scoring yang menghasilkan akurasi 90%. Sistem ini menggunakan 5 input fuzzy dan 10 input scoring, namun kompleksitas yang tinggi membuat sistem kurang efisien untuk implementasi real-time.

Mappeasse et al. (2022) fokus pada penentuan laptop bekas dengan 4 variabel fuzzy. Meskipun lebih sederhana, sistem ini terbatas pada domain laptop bekas dan belum mengakomodasi kebutuhan yang lebih luas.

Febrianto et al. (2024) menggunakan Fuzzy Database Model Tahani yang mampu menangani query ambigu. Namun, sistem ini memerlukan infrastruktur database yang kompleks dan kurang praktis untuk pengguna individual.

1. Spesifikasi Kebutuhan Sistem Baru

Berdasarkan analisis sistem terdahulu dan identifikasi masalah, dikembangkan spesifikasi kebutuhan untuk sistem baru yang dapat mengatasi keterbatasan-keterbatasan yang ada.

2. Kebutuhan Fungsional

Sistem harus mampu menerima input kriteria dari pengguna dengan interface yang intuitif. Input mencakup:

- a) Rentang harga yang diinginkan (dalam rupiah);
- b) Preferensi jenis prosesor (entry-level, mid-range, high-end);
- c) Kebutuhan kapasitas RAM (dalam GB);
- d) Kebutuhan kapasitas storage (dalam GB);
- e) Kebutuhan kemampuan VGA (integrated, dedicated entry, dedicated gaming);
- f) Tujuan penggunaan laptop (office, multimedia, gaming, professional).

3. Kebutuhan Perangkat

Kebutuhan Perangkat Keras:

- a) Processor minimal Intel i5 atau AMD Ryzen 5 untuk development
- b) RAM minimal 8GB untuk *smooth operation*
- c) Storage minimal 500GB SSD untuk *performance*

4. Kebutuhan Perangkat Lunak:

- a) Operating System: Windows 10/11, macOS, atau Linux
- b) Development Environment: Python 3.8+ dengan library fuzzy logic
- c) Database: PostgreSQL atau MySQL untuk data storage
- d) Testing Framework: pytest untuk *automated testing*

5. Arsitektur Sistem

- a. Presentation Layer : Layer ini bertanggung jawab untuk interaksi dengan pengguna melalui interface yang intuitif.
- b. Business Logic Layer : Layer ini mengimplementasikan algoritma Fuzzy Mamdani dan business rules;
- c. Data Access Layer : Layer ini mengelola akses ke data dan knowledge base;
- d. Data Layer : Layer ini menyimpan semua data yang diperlukan sistem

4.2.2 Perancangan Database

Database dirancang dengan struktur relational yang normalized untuk memastikan data integrity dan efficiency. Skema database utama meliputi:

1. Tabel Laptop:

- a. laptop_id (Primary Key)
- b. brand, model, series
- c. processor_type, processor_speed
- d. ram_capacity, ram_type
- e. storage_capacity, storage_type
- f. vga_type, vga_memory
- g. screen_size, resolution
- h. price, availability
- i. created_date, updated_date

2. Tabel Fuzzy_Rules:

- a. rule_id (Primary Key)
- b. rule_condition (IF part)
- c. rule_conclusion (THEN part)
- d. rule_weight, rule_priority
- e. is_active, created_date

3. Tabel User_Session:

- a. session_id (Primary Key)
- b. user_inputs (JSON format)
- c. system_outputs (JSON format)
- d. timestamp, session_duration

4.2.3 Perancangan Fuzzy Controller

Fuzzy Controller dirancang untuk mengimplementasikan metode Mamdani dengan struktur yang modular dan extensible:

- a) Fuzzification Module

Modul ini bertanggung jawab untuk mengkonversi input crisp menjadi nilai fuzzy:

```
def fuzzify(input_value, membership_functions):
    fuzzy_values = {}
    for mf_name, mf_params in membership_functions.items():
        fuzzy_values[mf_name] = calculate_membership(input_value, mf_params)
    return fuzzy_values
```

b) Rule Evaluation Module

Modul ini mengevaluasi setiap aturan fuzzy menggunakan operator MIN:

```
def evaluate_rules(fuzzy_inputs, rule_base):
    rule_outputs = {}
    for rule in rule_base:
        antecedent_value = min([fuzzy_inputs[var][term]
                                for var, term in rule.antecedent])
        rule_outputs[rule.id] = min(antecedent_value, rule.weight)
    return rule_outputs
```

c) Aggregation Module

Modul ini mengkonversi output fuzzy menjadi nilai crisp menggunakan metode Centroid:

```
def defuzzify(aggregated_output, output_range):
    numerator = sum(value * point for point, value in aggregated_output.items())
    denominator = sum(aggregated_output.values())
    return numerator / denominator if denominator != 0 else 0
```

Perancangan Interface

Interface dirancang dengan prinsip user-centered design untuk memastikan kemudahan penggunaan:

1. Main Dashboard

Overview sistem dengan quick access ke fitur utama

Recent recommendations dan user activity

System performance indicators

2. Input Form

Step-by-step wizard untuk input kriteria

Interactive sliders dan dropdown untuk user experience

Real-time validation dan feedback

Help tooltips untuk *guidance*

3. Results Display

Ranking recommendations dengan detailed explanation

Comparison matrix untuk alternative options

Reasoning visualization untuk transparency

Export functionality untuk further analysis

4. Analytics Dashboard

Usage statistics dan trends

Performance metrics visualization

User feedback dan satisfaction scores

System optimization recommendations

Perancangan sistem ini memastikan bahwa implementasi Fuzzy Mamdani dapat dilakukan dengan efektif sambil memberikan experience yang optimal bagi pengguna akhir.

4.2.4 Alur Klasifikasi Laptop

Berikut adalah proses klasifikasi laptop memakai Metode Fuzzy Mamdani yang mengacu pada sistem SPK dalam penentuan laptop:

1. Input Data

Pengguna atau sistem menginput data mengenai spesifikasi atau karakteristik laptop:

- a) Harga
- b) Prosesor (kecepatan, tipe)
- c) RAM
- d) Kapasitas Hardisk/Storage

2. Fuzzifikasi

Setiap input yang disebutkan sebelumnya diganti berupa nilai tingkatan keanggotaan dalam himpunan fuzzy, seperti:

- a) Nilai RAM 4GB akan dianalisis untuk menentukan sejauh mana ia tergolong dalam kategori "rendah", "sedang", atau "tinggi".

b) Setiap variabel memanfaatkan fungsi keanggotaan, seperti: rendah, sedang, tinggi untuk RAM dan harga.

3. Rule Base

Aturan IF-THEN disusun berdasar pengetahuan atau kebutuhan yang ada, seperti:

- a) Jika RAM besar DAN Harga terjangkau, MAKA Laptop sangat dianjurkan.
- b) Peraturan ini mengaitkan semua kemungkinan gabungan dari masukan.

4. Penarikan Kesimpulan

Proses pengambilan keputusan dengan logika fuzzy dilakukan:

- a) Untuk setiap ketentuan:

Diperlukan nilai terendah (operator AND – fungsi t-norm, umumnya adalah MIN) dari derajat keanggotaan input dalam aturan.

- b) Mengintegrasikan hasil semua aturan dengan menggunakan nilai tertinggi (operator OR – t-conorm, umumnya MAX) pada setiap keluaran.

5. Proses Defuzzifikasi

Output yang tidak jelas (grafik/area linguistik seperti “direkomendasikan”, “kurang direkomendasikan”, “tidak direkomendasikan”) diubah menjadi nilai yang jelas (angka pasti).

- a) Metode yang umum digunakan adalah Centroid.
- b) Nilai crisp ini menggambarkan sejauh mana setiap pilihan laptop sesuai dengan kebutuhan pengguna.

6. Saran dan Penilaian

Sistem akan menunjukkan hasil akhir yang berbentuk:

- a) Daftar atau rangkaian laptop berdasarkan tingkat kecocokan dari yang tertinggi hingga terendah sesuai dengan kebutuhan pengguna.
- b) Bisa juga dalam bentuk kategori seperti “Sangat Dianjurkan”, “Dianjurkan”, “Kurang Dianjurkan”.

4.3.1 Flowchart

Flowchart menggambarkan langkah-langkah alur kerja sistem secara urut dari awal hingga akhir.

