

1 Pendahuluan

1.1 Latar belakang

Berdasarkan cara membangun solusi, sistem pakar terdiri atas dua jenis yakni sistem pakar klasifikasi dan sistem pakar konstruksi [8]. Kedua sistem pakar tersebut memiliki perbedaan, terutama pada pembangunan solusi yang dihasilkan. Pada sistem pakar klasifikasi, solusi sudah didefinisikan sebelumnya tinggal dipilih sesuai spesifikasi dari user. Sedangkan sistem pakar konstruksi masih harus menyusun solusi dari elemen-elemen pembangun sesuai *requirement* dan *constraint* user.

Ada beberapa karakteristik permasalahan yang cocok diselesaikan oleh masing-masing sistem pakar. Untuk sistem pakar konstruksi, permasalahan yang diselesaikan lebih kepada penyusunan komponen/elemen, misalkan sistem pakar konstruksi menu makanan (komponen terdiri dari jadwal dan jenis makanan), sistem pakar konstruksi bangunan (komponen terdiri dari atap, dinding, rangka), dan sistem pakar konstruksi pakaian (komponen berupa atasan, bawahan, aksesoris). Sistem pakar klasifikasi cocok untuk permasalahan yang sudah ditentukan himpunan solusinya. Contohnya sistem pakar klasifikasi kredit (solusi berupa kredit diterima, diterima dengan syarat, ditolak), sistem pakar klasifikasi bangunan (solusi berupa rumah tipe A, tipe B, tipe C), dan sistem pakar klasifikasi perkiraan cuaca (solusi berupa cerah, berawan, hujan, dan lain-lain).

Pembangunan sistem pakar yang ada pada umumnya hanya menangani satu domain persoalan saja, sehingga pembangunan sistem pakar untuk persoalan yang beragam harus selalu dimulai dari tahap awal. Akan sangat efisien proses pembangunan sistem pakar manakala aspek perangkat lunak bisa di *reuse* dari perangkat lunak sebelumnya dengan cara mengubah *setting* atau memcustomisasi kemudian mengisinya dengan pengetahuan berdasarkan konteks domain permasalahan yang sedang diangkat. Dengan cara ini, *knowledge engineer* cukup menambahkan pengetahuan dari domain masalah sistem pakar tanpa harus merekonstruksi perangkat lunak itu sendiri. Hal ini yang disebut dengan *shell* atau *engine* sistem pakar.

Engine sistem pakar yang dibangun yaitu *engine* sistem pakar konstruksi dengan menggunakan *configuration design problem solving*. Persoalan konfigurasi yang diselesaikan berjenis *skeletal design* menggunakan metoda *hierarchical configuration*. Pakar dapat mengubah susunan *skeletal plan* sesuai domain persoalan yang diselesaikan. Solusi yang *fixed* akan didapatkan setelah *skeletal plan* diproses dengan melihat *requirement* dan *constraint* yang ada.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan yang akan diteliti dan dianalisis yaitu sebagai berikut :

1. Bagaimana menerapkan *engine* pada beberapa masalah kepakaran yang solusinya sesuai dengan solusi pakar yang sudah ada.
2. Bagaimana membangun editor basis pengetahuan dan *skeletal plan* yang bisa mendefinisikan domain masalah secara fleksibel.
3. Bagaimana membuat antarmuka (interface) interaksi konsultasi user dengan sistem pakar yang sesuai dengan permasalahan masing-masing.

Batasan masalah dalam pembangunan *engine* untuk sistem pakar konstruksi ini adalah sebagai berikut :

1. Pengujian pada *engine* ini dibatasi dalam dua domain persoalan dengan *skeletal design* berbeda, yaitu sistem pakar konstruksi untuk diet golongan darah dan komposisi obat.
2. *Engine* ini hanya dapat menangani sistem pakar konstruksi dengan configuration design problem solving yang berjenis skeletal design dan menggunakan metode hierarchical configuration.
3. *Engine* ini hanya dikonfigurasi oleh *administrator (knowledge engineer)* dan user biasa hanya bisa menggunakan sistem pakar konstruksi yang sudah ada.
4. Admin yang membangun komponen dan *knowledge engine* ini diasumsikan sudah memahami pengetahuan tentang sistem pakar konstruksi yang akan dibuat.
5. Untuk mendapatkan solusi, diasumsikan semua *knowledge, requirement, constraint* dan komponen yang ada sudah terisi atau dipenuhi oleh admin.

1.3 Tujuan

Tujuan dari pembangunan *engine* sistem pakar konstruksi ini adalah sebagai berikut :

1. Menyediakan fasilitas bagi user untuk membangun sistem pakar untuk domain permasalahan berbeda.
2. Menganalisis usability dari editor basis pengetahuan untuk digunakan pada berbagai domain persoalan yang bisa ditangani engine.
3. Menganalisis fleksibilitas skeletal plan dari sisi kedalaman level, cabang, dan posisi AND OR.
4. Membangun antarmuka (interface) interaksi konsultasi user dengan sistem pakar yang sesuai dengan permasalahan masing-masing.
5. Menganalisis solusi yang dihasilkan engine menurut kebenaran sesuai hasil dari pakar yang sudah ada dan kemudahan memahami solusi.

1.4 Metodologi penyelesaian masalah

Metode penyelesaian masalah pada Tugas Akhir ini adalah :

1. Studi Literatur
Yaitu mencari referensi-referensi, mempelajari, dan mendalami materi yang berhubungan sistem informasi dan sistem pakar.
2. Melakukan metoda *Expert System Development Life Cycle (ESDLC)*
 - a. Identifikasi masalah yang akan ditangani sistem pakar konstruksi (*problem identification*)
 - b. Analisis kebutuhan dan akuisisi pengetahuan (*preliminary requirement analysis and knowledge acquisition*)
 - c. Pemilihan tools sistem pakar (*selection of tools*)
 - d. Representasi pengetahuan (*representation*)
 - e. Verifikasi dan validasi (*verification and validation*)
 - f. Implementasi (*implementation*)
 - g. Pengoperasian dan pemeliharaan (*operation and maintenance*)
3. Penyusunan laporan tugas akhir