

APLIKASI SHELL SISTEM PAKAR

Yeni Agus Nurhuda¹, Sri Hartati²

¹Program Studi Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Teknokrat Lampung
Jl. Z.A. Pagar Alam 9-11 Labuhan Ratu, Kedaton, Bandar Lampung
Telp. (0721)702022 Fax (0721)702022

²Magister Ilmu Komputer Universitas Gadjah Mada Yogyakarta
E-mail: nurhuda5875@yahoo.com

ABSTRAK

Penelitian ini dilakukan atas dasar kebutuhan akan adanya shell sistem pakar yang dapat digunakan sebagai alat untuk mengembangkan sistem pakar oleh para knowledge engineer atau bahkan seorang pakar dengan efisien. Dengan sistem ini diharapkan seorang knowledge engineer atau seorang pakar dapat dengan mudah mengembangkan sistem pakar dengan waktu dan biaya yang sangat murah. Pada dasarnya shell sistem pakar merupakan sistem pakar yang tanpa dilengkapi dengan basis pengetahuan. Shell sistem pakar ini terdiri atas 2 bagian utama, yaitu bagian pengembangan sistem pakar, dan bagian konsultasi. Proses inferensi pada sistem ini menggunakan metode forward chaining. Mekanisme inferensi dilakukan dengan melakukan penelusuran basis pengetahuan yang direpresentasikan dalam pohon inferensi. Proses pengambilan keputusan dilakukan berdasarkan aturan dengan menelusuri pohon inferensi dari akar hingga node terakhir yang merupakan keputusan/kesimpulan dari aturan yang ada. Basis pengetahuan dari sistem ini terdiri atas fakta (kondisi), kesimpulan, dan aturan pada pohon inferensi. Fakta dan kesimpulan dapat dilengkapi dengan gambar grafis pada bagian penjelasan. Keluaran dari shell ini adalah sistem pakar yang akan dapat memberikan kesimpulan dan penjelasan berdasarkan fakta-fakta yang diberikan selama proses konsultasi.

Kata Kunci: Sistem Pakar, Knowledge Engineer, Forward Chaining, Pohon Inferensi

1. PENDAHULUAN

Sistem pakar merupakan sistem yang berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Komponen utama dari sebuah sistem pakar adalah basis pengetahuan dan mesin inferensi (Ignizio, 1991). Basis pengetahuan berisi fakta-fakta dan kaidah-kaidah, sedangkan mesin inferensi merupakan metode penalaran terhadap permasalahan sebagaimana dilakukan oleh seorang pakar dalam menalar permasalahan di bidangnya.

Pembentukan sistem yang berfikir seperti manusia membutuhkan basis pengetahuan. Dalam sistem pakar, basis pengetahuan adalah inti keseluruhan sistem karena merupakan representasi pengetahuan (*knowledge representataion*) dari seorang pakar.

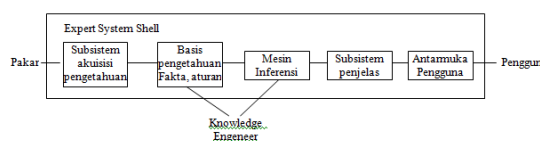
Sudah banyak penelitian tentang sistem pakar dan implementasinya, tetapi dari sekian banyak sistem pakar yang dikembangkan kebanyakan dibuat hanya untuk menangani permasalahan tertentu. Penelitian tentang generator sistem pakar (*shell*) masih sangat sedikit, padahal perbedaan sistem pakar yang satu dengan yang lainnya hanya terletak pada pangkalan pengetahuannya, mesin inferensi pangkalan data dan user interface akan dapat bekerja dengan semua pangkalan pengetahuan.

Dalam banyak penelitian, komunitas AI lebih memilih menggunakan bahasa konvensional atau bahasa pengembangan sistem pakar yang biasa disebut bahasa AI seperti LISP dan Prolog untuk mengembangkan aplikasi sistem pakar.

Kecenderungan tidak dikembangkannya shell sistem pakar karena alasan keterbatasan yang akan muncul, dianggap tidak berguna dan cenderung dianggap sebagai mainan. Cukup aneh, padahal opini yang telah berkembang adalah **“jika sesuatu dapat dikembangkan dengan biaya yang murah, mudah digunakan dan mudah dikembangkan, maka ini harus diseriusi”**. Kondisi inilah yang mendorong dilakukannya penelitian ini.

1.1 Shell Sistem Pakar (*Expert System Shell*)

Expert System Shell adalah paket perangkat lunak yang memfasilitasi pembangunan sistem berbasis pengetahuan dengan menyediakan sekema representasi pengetahuan dan sebuah mesin inferensi. Pada kenyataannya *Expert System Shell* adalah arsitektur umum dari sistem pakar tanpa domain pengetahuan spesifik dari pakar, sehingga *shell* dapat diisi dengan pengetahuan yang berbeda-beda untuk aplikasi yang berbeda. Diasosiasikan sebagai *shell* karena merupakan metode yang digunakan pengembangan aplikasi dengan mengkonfigurasi dan melengkapi komponen-komponen yang ada didalamnya. Beberapa komponen dasar shell sistem pakar ditunjukkan pada gambar 1. berikut:

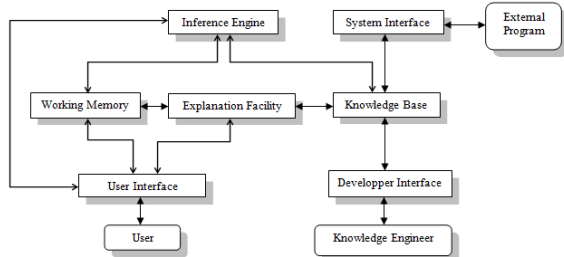


Gambar 1. Komponen Dasar *Shell* Sistem Pakar (Nii., 1993)

- a. Basis pengetahuan menyimpan fakta dan aturan. Sebuah *expert system tool* memiliki satu atau lebih skema representasi pengetahuan untuk mengekspresikan pengetahuan tentang domain aplikasi.
- b. Mesin inferensi menyediakan mekanisme inferensi untuk memanipulasi informasi simbolik dan pengetahuan dalam basis pengetahuan untuk melakukan penelusuran dalam menyelesaikan masalah.
- c. Subsistem akuisisi pengetahuan merupakan subsistem untuk membantu pakar dalam membangun basis pengetahuan. Pengumpulan pengetahuan diperlukan untuk menyelesaikan masalah dan membangun basis pengetahuan lebih jauh untuk meningkatkan kemampuan sistem pakar.
- d. Subsistem penjelasan merupakan penjelas yang menjelaskan tentang aksi sistem. Penjelasan dapat berupa bagaimana kesimpulan akhir diperoleh dan dari mana solusi yang diberikan didapatkan.
- e. Antar muka pengguna merupakan sarana komunikasi antara sistem dengan pengguna.

1.2 Struktur Shell sistem Pakar

Berdasarkan komponen dasar penyusunnya, struktur *shell* sistem pakar secara umum tidak berbeda dengan arsitektur sistem pakar. Struktur *shell* sistem pakar ditunjukkan dalam gambar 2.



Gambar 2. Arsitektur Shell Sistem Pakar (Darkin, 1998)

- a. **Knowledge Base** memuat pengetahuan khusus tentang wilayah permasalahan yang diambil dari pakar yang meliputi fakta masalah, aturan, konsep, dan keterkaitannya.
- b. **Inference Engine** adalah pemroses pengetahuan (merupakan program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan dan blackboard, serta digunakan untuk memformulasikan pengetahuan).
- c. **Working Memory** memuat fakta-fakta yang dimasukkan oleh pengguna atau hasil inferensi pakar selama proses konsultasi.
- d. **User Interface** adalah sarana penghubung antara pengguna dengan sistem.
- e. **Developer Interface** adalah sarana penghubung knowledge engineer untuk mengembangkan sistem pakar.

- f. **Explanation Facility** memberikan penjelasan (kepada pengguna mengenai hasil konsultasi. Fasilitas penjelasan diberikan untuk menjelaskan bagaimana proses penarikan kesimpulan, biasanya dilakukan dengan memperlihatkan aturan yang digunakan.)
- g. **System Interface** menghubungkan sistem pakar dengan program luar seperti basis data, spreadsheet, algoritma dan lain-lain.

1.3 Proses Inferensi

Proses penggabungan banyak aturan berdasarkan data yang tersedia disebut inferensi. Diperlukan program komputer untuk mengakses pengetahuan dalam proses inferensi. Program ini adalah algoritma yang mengontrol proses pertimbangan dan disebut sebagai mesin inferensi atau program kontrol. Dalam sistem berbasis aturan, disebut juga penerjemah aturan (Turban *et al*, 2005).

Program kontrol yang paling populer untuk sistem berbasis-aturan, yaitu *forward chaining* dan *backward chaining*.

- a. **Forward chaining** adalah pendekatan *data-driven*. Investigasi dimulai dari informasi yang tersedia atau dari ide dasar, kemudian dicoba menarik kesimpulan. Pencocokan fakta dilakukan dengan mencari bagian JIKA (IF) terlebih dahulu. Setelah semua bagian JIKA (IF) terpenuhi, aturan dipilih untuk mendapatkan kesimpulan. Jika kesimpulan yang diambil dari keadaan pertama, bukan dari yang terakhir maka ia akan digunakan sebagai fakta untuk disesuaikan dengan kondisi JIKA. Dengan kata lain penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.
- b. **Backward Chaining** adalah pendekatan *goal-driven*, investigasi dimulai dari harapan apa yang akan terjadi (hipotesis), kemudian mencari bukti yang mendukung (atau berlawanan) dengan harapan. Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (MAKA). Dengan kata lain, penalaran dimulai dari hipotesis dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta yang ada dalam basis pengetahuan.
- c. **Pohon Inferensi** (disebut juga pohon tujuan atau pohon logika) menyediakan tampilan skematis proses inferensi. Pohon inferensi serupa dengan pohon keputusan dan diagram pengaruh (Turban, 2002). Dalam membangun pohon inferensi, premis dan kesimpulan ditunjukkan sebagai node. Cabang menghubungkan premis dan kesimpulan. Operator AND dan OR digunakan untuk merefleksikan struktur aturan.

2. RANCANGAN SHELL

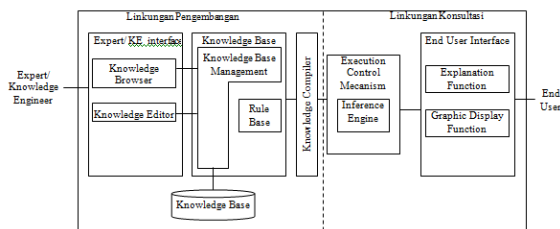
Perangkat lunak yang dikembangkan pada penelitian ini adalah *Shell* sistem pakar (*Expert System Shell*). Perangkat lunak ini nantinya dapat digunakan dalam dua fungsi utama, yaitu fungsi

sebagai sistem pakar dan fungsi sebagai perangkat lunak untuk mengembangkan sistem pakar.

Shell sistem pakar yang dikembangkan dalam penelitian ini memiliki dua bagian pokok, yaitu lingkungan pengembangan dan lingkungan konsultasi. Pengguna dari sistem ini dibedakan dalam 2 kelompok pengguna, masing-masing adalah pengguna dalam kelompok pengembang sistem pakar dan pengguna dalam kelompok pemakai sistem pakar (penerima manfaat).

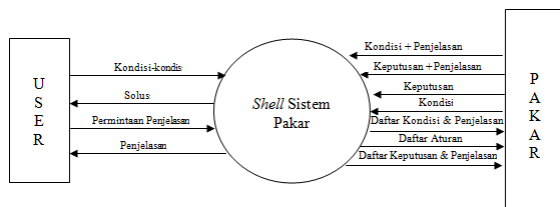
2.1 Arsitektur Shell

Arsitektur *shell* sistem pakar yang dikembangkan dalam penelitian ini merupakan hasil modifikasi dari struktur *shell* sistem pakar ES/KERNEL2 yang telah dikembangkan oleh Hitachi.Ltd. tahun 1991 (gambar 3).



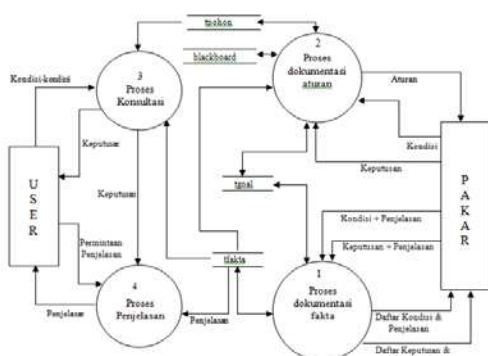
Gambar 3. Struktur *Shell* Sistem Pakar Dimodifikasi dari struktur *shell* ES/KERNEL2 Hitachi.Ltd (Nii, 1993)

Gambaran umum arus data dalam *shell* sistem pakar yang dikembangkan dapat dilihat pada gambar 4.



Gambar 4. Diagram Arus Data Level 0 *Shell* Sistem Pakar

Terdapat 4 proses utama dalam *shell* sistem pakar yang dikembangkan (dapat dilihat dalam gambar 4). DFD level 2 dapat dilihat pada gambar 3.4 dan 3.5. (Nurhuda, 2009).



Gambar 5. Diagram Arus Data Level 1

2.2 Representasi Pengetahuan

Basis pengetahuan dalam *shell* sistem pakar yang dikembangkan ini direpresentasikan dalam bentuk pohon inferensi. Representasi ini dipilih dengan pertimbangan bahwa representasi pohon merupakan representasi pengetahuan yang fleksibel dan dapat dengan mudah dirubah ke dalam bentuk aturan, begitu juga sebaliknya. Konversi dilakukan secara otomatis sebelum dilakukan penalaran oleh mesin inferensi sehingga representasi pengetahuan menjadi lebih fleksibel.

2.3 Mesin Infrensi (*Inference Engine*)

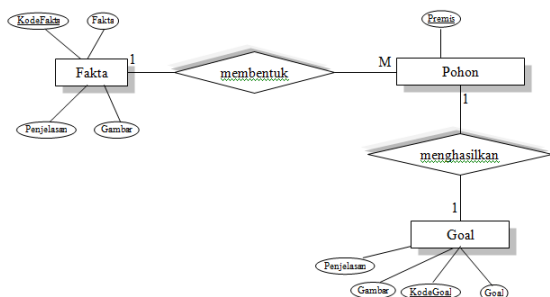
Mekanisme inferensi pada *shell* yang dikembangkan ini dilaksanakan dengan menelusuri jalur/*link* pada pohon inferensi, dimulai dari root (keadaan awal) mengikuti jalur/*link* berdasarkan fakta baru yang diperoleh. Secara garis besar, sistem pelacakan untuk mendapatkan keputusan dalam *shell* sistem pakar yang dikembangkan ini menggunakan tahapan inferensi berdasarkan data input berupa kondisi-kondisi yang diperoleh selama proses konsultasi yang dilaksanakan dengan menelusuri arah link/jalur pada pohon inferensi. Mesin inferensi untuk merunut dan mengeksplorasi basis pengetahuan yang digunakan dalam *shell* yang dikembangkan adalah menggunakan sistem pelacakan maju (*forward chaining*).

Metode pelacakan maju (*forward chaining*) dalam *shell* ini digunakan dengan pertimbangan bahwa dalam proses konsultasi, pengguna akan dibawa pada titik kesimpulan berdasarkan fakta-fakta yang dimasukkan dengan menelusuri pohon inferensi. Tidak ada hipotesis awal dari permasalahan yang dikonsultasikan, akan tetapi fakta yang dimunculkan yang akan dieksplorasi berdasarkan pohon inferensi untuk mendapatkan kesimpulan akhir.

2.4 Basis Pengetahuan

Basis pengetahuan dalam *shell* sistem pakar yang dikembangkan berisi fakta dan aturan. Fakta dalam basis pengetahuan dari *shell* sistem pakar ini dikelompokkan menjadi 3 bagian, yaitu bagian yang mendokumentasikan fakta, bagian yang mendokumentasikan kesimpulan, dan bagian yang mendokumentasikan penjelasan. Aturan dalam basis pengetahuan *shell* sistem pakar yang dikembangkan ini didokumentasikan dan direpresentasikan dalam bentuk pohon inferensi.

Fakta dan aturan dalam basis pengetahuan *shell* sistem pakar yang dikembangkan ini didokumentasikan dalam basis data relasional yang terdiri atas table-table yang memiliki hubungan saling ketergantungan. Masing-masing tabel dan hubungan antara satu tabel dengan tabel yang lain digambarkan dalam bentuk diagram hubungan antar entitas (*entity relationship diagram*) dalam gambar 6.



Gambar 6. . Diagram E-R (Entity Relationship Diagram) shell sistem pakar

3. IMPLEMENTASI DAN HASIL

Penelitian menghasilkan sebuah *shell* sistem pakar yang dapat digunakan untuk mengembangkan aplikasi sistem pakar. *Shell* sistem pakar yang dihasilkan memiliki 2 fungsi utama, fungsi untuk pengembangan sistem pakar dan fungsi untuk konsultasi dari sistem pakar yang dihasilkan.

Fungsi pengembangan sistem pakar digunakan oleh pakar atau *knowledge engineer* untuk mengembangkan aplikasi sistem pakar. Fasilitas untuk meembangkan sistem pakar pada *shell* ini terdiri atas fasilitas untuk mendokumentasikan fakta/kondisi, fasilitas untuk mendokumentasikan kesimpulan, dan fasilitas untuk mendokumentasikan aturan. Contoh dari dokumentasi fakta/kondisi diberikan pada gambar 7 dan 8, contoh untuk dokumentasi kesimpulan diberikan pada gambar 9 dan 10, dan contoh untuk dokumentasi aturan diberikan pada gambar 11.



Gambar 7. From dokumnetasi fakta/kondisi



Gambar 8. Daftar fakta/kondisi pada dokumentasi fakta/kondisi



Gambar 9. Form dokumentasi kesimpulan



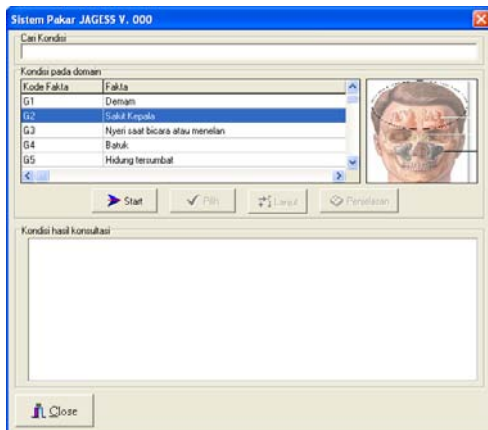
Gambar 10. Form daftar kasimpulan terdokumentasi



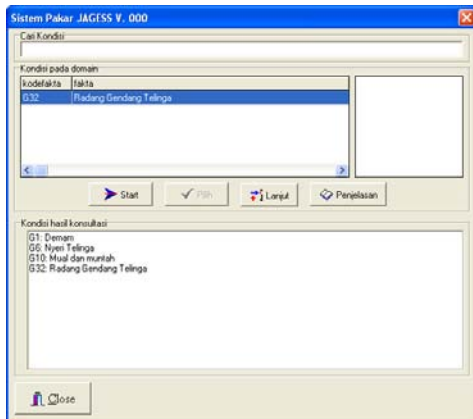
Gambar 11. Form dokumentasi aturan

Hasil pengembangan sistem pakar dengan menggunakan *shell* yang dikembagkan ini adalah sistem pakar yang dapat digunakan oleh pengguna untuk berkonsultasi mengenai domain masalah yang telah dibangun basis pengetahuannya oleh pengembang (pakar/*knowledge engineer*). Fasilitas bagi pengguna sistem pakar terdiri atas fasilitas konsultasi dan kesimpulan hasil konsultasi. Untuk berkonsultasi dengan sistem pakar, pengguna harus memilih salah satu kondisi yang sesuai dengan keadaan yang dimunculkan sistem. Sebagai hasil konsultasi adalah kesimpulan yang dilengkapi dengan penjelasan mengenai kesimpulan itu. Contoh

konsultasi dan hasil konsultasi dari sistem pakar digambarkan pada gambar 12, 13 dan 14.



Gambar 12. Awal konsultasi dengan sistem pakar



Gambar 13. Form Konsultasi dengan sistem pakar



Gambar 14. Hasil konsultasi dengan sistem pakar

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil penelitian pengembangan *shell* sistem pakar yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

- Prototype shell* sistem pakar yang dikembangkan menggunakan pola inferensi menelusuri pohon inferensi yang dikembangkan dengan model pohon keputusan.
- Pola konsultasi pada sistem pakar yang dihasilkan dilakukan dengan meminta pengguna untuk memilih fakta kondisi yang dimunculkan berdasarkan hasil penelusuran pohon inferensi.

- Dalam aplikasi ini aturan baru dapat ditambahkan tanpa mempengaruhi mesin inferensi.
- Shell* sistem pakar dapat digunakan untuk mengembangkan sistem pakar dengan memasukkan basis pengetahuan tanpa harus melalui proses pengkodean pada bahasa pemrograman.

4.2 Saran

Penelitian lebih lanjut perlu dilakukan terutama dalam hal:

- Pengujian *shell* lebih lanjut dengan basis pengetahuan pada domain masalah yang lain.
- Pengembangan *shell* sistem pakar dengan mempertimbangkan faktor ketidakpastian.

PUSTAKA

Darkin, J., 1998, *The Handbook of Applied Expert System*, Google Book. Diakses lewat <http://www.books.google.co.id>

Ignizio, P.J., 1991, *Introduction to Expert Systems: The Development and Implementation of Rule-Based Ekspert Systems*, McGraw-Hill, Inc, USA

Nii, H.P., 1993, *Tools and Infrastructure for Knowledge-Based Systems*, WTEC Hyper Lybrarian, Diakses lewat http://www.wtec.org/loyola/kb/c3_s1.htm.

Nurhuda, Y.A., 2009, *Rancang Bangun Shell Sistem Pakar*, Tesis S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta

Turban, E, Aronson, Jay E., Liang, Ting-Peng, 2005, *Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)*, Penerbit Andi, Yogyakarta