

Intelligent Tutoring System: Expert-Knowledge Module Menggunakan Case-Based Reasoning

Abdiansah¹

¹Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya
Jln. Raya Palembang Prabumulih, Km.32 Indralaya, Ogan Ilir, Indonesia, 30662
e-mail: ¹abdiansah@unsri.ac.id

Abstrak

Modul sistem pakar telah dikembangkan untuk intelligent tutoring system berbasis case-based reasoning (CBR) dalam ranah pemrograman komputer. Sistem ini bertujuan untuk membantu pelajar dalam mempelajari bahasa pemrograman terutama praktik pemrograman karena sistem yang dibuat dilengkapi modul untuk mencari solusi bagi pesan-pesan kesalahan yang muncul. Data kasus untuk CBR diambil dari kasus-kasus kesalahan yang terjadi pada saat pemrograman. Pengujian menunjukkan bahwa tingkat akurasi sistem tergantung dari banyaknya kasus-kasus yang tersimpan dalam basis data kasus. Semakin banyak kasus yang tersimpan, maka tingkat akurasi sistem akan semakin meningkat.

Kata kunci: *modul sistem pakar, intelligent tutoring system, case-based reasoning.*

Abstract

Expert system module have been developed for intelligent tutoring system based on case-based reasoning (CBR) in computer programming field. This system purposes to help the students in studying the programming language especially programming practice because the system has completed by module to find solution for error messages that appeared. Data case for CBR were taken from error cases that happened in programming. Testing process showed that degree of system's accuracy depend on number of cases in database case. The degree of system accuracy would be in creasing to follow the more and more saving cases.

Keywords: *expert system module, intelligent tutoring system, case-based reasoning.*

1. Pendahuluan

Intelligent Tutoring System (ITS) merupakan program komputer yang dirancang untuk menggabungkan teknik-teknik yang berasal dari komunitas *Artificial Intelligence (AI)* untuk membuat tutor/pengajar yang mengetahui apa yang mereka ajar (*what they teach*), siapa yang mereka ajar (*who they teach*) dan bagaimana cara mereka mengajar (*how they teach*) [1]. Menurut [2] tujuan dari ITS adalah memungkinkan bagi pelajar untuk memperoleh pengetahuan dan mengembangkan kemampuan dalam suatu bidang tertentu. Untuk mendapatkan pengajaran yang efektif, sistem ITS harus dilengkapi dengan representasi pengetahuan yang komprehensif dari bidang yang menjadi pokok pengajaran. Selain itu, ITS juga harus dilengkapi dengan mekanisme yang dapat menggunakan representasi pengetahuan tersebut untuk melakukan penalaran (*reasoning*) dalam memecahkan suatu permasalahan (*problem-solving*).

Usaha pertama kali pembuatan aplikasi komputer untuk pengajaran lahir pada tahun 1960-an dan 1970-an yang dipelopori [4] dimana istilah yang digunakan pada saat itu disebut dengan *Intelligent Computer-Assisted Learning* (ICAI). Secara umum terdapat empat komponen penyusun ITS yaitu: 1) Modul Pengetahuan Pakar (*The Expert Knowledge Module*); 2) Modul Model Pelajar (*The Student Model Module*); 3) Modul Pengajar (*The Tutor Module*) dan 4) Modul Antarmuka Pengguna (*The User Interface Module*). Berdasarkan literatur yang diambil dari [3] sudah banyak penelitian-penelitian yang membahas ke-empat komponen ITS tersebut. Salah satu komponen yang cukup vital dalam ITS adalah komponen modul pengetahuan pakar. Komponen ini berisi pengetahuan seorang pakar dari suatu bidang yang berisi konsep, aturan-aturan dan strategi pemecahan masalah. Akuisisi dan kodifikasi pengetahuan merupakan pekerjaan yang sangat sulit dan memakan waktu lama terutama untuk bidang yang kompleksitasnya tinggi [1]. Oleh karena itu komponen ini menjadi *central issue* dalam ITS dan dibutuhkan penelitian yang lebih mendalam lagi. Terdapat dua fungsi utama dari modul ini yaitu: 1) sebagai sumber pengetahuan yang akan diberikan untuk pelajar, seperti membuat, menjelaskan dan merespon suatu pertanyaan dan; 2) memberikan standar untuk mengevaluasi kemampuan pelajar [1].

Representasi pengetahuan dalam modul pengetahuan pakar dapat dibuat menggunakan beberapa metode seperti *semantic network*, *frames*, *production rules* dan lainnya. Penelitian ini menggunakan *case-based reasoning* (CBR) sebagai representasi pengetahuan sekaligus sebagai mekanisme penalaran bagi komponen ITS. CBR diperkenalkan oleh Winston yang dikembangkan dari sistem pembelajaran berbasis kesamaan (*similarity-based learning*). Secara sederhana CBR merupakan sebuah sistem yang menggunakan pengalaman lama untuk dapat mengerti dan menyelesaikan masalah baru [5][6]. CBR memiliki kemudahan dalam representasi kasus karena hanya menggunakan dua komponen yaitu *problem* dan *solution*. Bagian *problem* berisi seluruh permasalahan dan bagian *solution* berisi seluruh penyelesaian. Penggunaan CBR diharapkan komponen modul pengetahuan pakar dapat dibangun dengan lebih mudah dan lebih cepat sehingga dapat memberikan nilai tambah dalam hal waktu pembuatan ITS secara keseluruhan. Oleh karena itu, penelitian ini mencoba untuk menggunakan CBR dalam membangun komponen modul pengetahuan pakar.

2. Metodologi Penelitian

2.1 Analisis Permasalahan

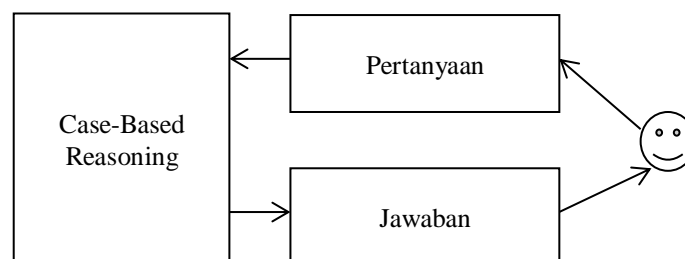
ITS merupakan bidang penelitian yang sangat luas dan beragam serta terus menerus dikembangkan. Keempat modul ITS: *Expert Knowledge*, *Student Model*, *Tutoring/Pedagogic* dan *User Interface*, merupakan bidang inti dalam penelitian ITS. Masing-masing modul dapat digunakan berbagai macam metode dan teknik serta dimungkinkan pula untuk melakukan hibridasi metode guna mencapai hasil yang maksimal. *Expert Knowledge Module* (EKM) menjadi salah satu inti dari penelitian ITS memberikan pengaruh yang cukup signifikan terhadap keberhasilan ITS. Salah satu fungsi dari MPP adalah menyediakan seperangkat pengetahuan yang diambil dari pakar dengan menggunakan suatu teknik tertentu serta mampu secara mandiri untuk menganalisis dan memberikan solusi terhadap suatu permasalahan. Berhasil tidaknya

suatu ITS dalam memberikan solusi terhadap suatu permasalahan tergantung dari bagus tidaknya rancangan dan metode yang digunakan dalam EKM.

CBR merupakan salah satu teknik dalam bidang kecerdasan buatan yang menggunakan analogi kasus dalam memecahkan suatu permasalahan. Dalam CBR, pengetahuan pakar direpresentasikan dalam bentuk kasus-kasus dan untuk mencari solusi terhadap kasus baru digunakan fungsi kemiripan. Sehingga berdasarkan kemampuan tersebut CBR dapat digunakan untuk membuat EKM. Oleh karena itu, permasalahan yang diangkat dalam penelitian ini adalah bagaimana cara mengembangkan perangkat lunak yang akan digunakan untuk menguji dan mengevaluasi metode CBR dalam membangun EKM sebagai komponen dari ITS. Domain permasalahan yang digunakan dalam penelitian ini adalah permasalahan di bidang pembelajaran pemrograman komputer. Salah satu alasan mengapa digunakan domain ini karena peneliti memiliki kemampuan cukup di bidang pemrograman komputer, sehingga akan mengurangi tingkat kesulitan pada tahap akuisisi pengetahuan. Secara konsep setiap domain permasalahan memiliki perbedaan dalam memecahkan masalah, tetapi sudut pandang metode hanya terdapat satu cara untuk memecahkan masalah yang sesuai dengan kaidah yang digunakan metode tersebut. Dengan kata lain, walaupun domain permasalahan berbeda-beda tetapi suatu metode memiliki satu cara untuk mencari solusi permasalahan.

Masalah utama yang sering dihadapi pelajar ketika praktik bahasa pemrograman adalah menemui kesalahan pada saat pengkodean program. Ada dua jenis pesan kesalahan dalam pengkodean yaitu kesalahan sintaks dan kesalahan logika. Kesalahan sintaks merupakan kesalahan dalam hal penulisan kode-kode dan struktur dari suatu bahasa pemrograman, sedangkan kesalahan logika adalah kesalahan dari logika yang dibuat oleh pemrogram. Umumnya *compiler* hanya bisa mendeteksi kesalahan sintaks, sedangkan untuk kesalahan logika harus diatasi sendiri oleh pemrogram. Pesan kesalahan sintaks yang ditampilkan oleh *compiler* berbahasa Inggris serta singkat, padat dan tidak bertele-tele. Terkadang pesan kesalahan tersebut sulit dimengerti oleh pemrogram ahli, terlebih lagi untuk pemrogram awam. Oleh karena itu, penelitian ini mencoba untuk membantu pemrogram dalam memahami pesan kesalahan yang terjadi pada saat memprogram. Setiap pesan kesalahan yang muncul diproses oleh CBR dan akan dicari solusi (dalam bahasa Indonesia) dari pesan kesalahan tersebut dengan disertai contoh kode program yang salah dan yang benar.

2.2 Arsitektur Sistem



Gambar 1: Arsitektur Sistem

Pada Gambar 1 dapat dilihat arsitektur sistem yang dibuat pada penelitian ini. Pengguna memberikan pertanyaan kepada sistem (CBR), pertanyaan yang diberikan diambil dari pesan kesalahan hasil dari *compiler* bahasa pemrograman. Sistem merespon dan memproses pertanyaan untuk dicari solusi yang tepat kemudian memberikan solusi tersebut kepada pengguna.

2.3 Analisis Data

Jenis data yang digunakan dalam penelitian ini merupakan data primer, dimana data dikumpulkan secara langsung dengan melihat pesan-pesan kesalahan yang terjadi dari beberapa kode program yang dibuat, bahasa pemrograman yang digunakan adalah bahasa Pascal. Tahap berikutnya adalah melakukan pengelompokan pesan, hal ini dilakukan karena ada beberapa pesan yang memiliki pola yang sama dan hanya beberapa kata yang berbeda. Sehingga total kelompok pesan kesalahan yang didapat berjumlah 22 pesan kesalahan.

Data pesan kesalahan yang didapat dibuat menjadi kasus CBR menggunakan teknik *frame* dan ditambah atribut-atribut penjelas lainnya. Terdapat 5 atribut kasus CBR yaitu: id, error, prog_salah, prog_benar dan solusi. Atribut id digunakan sebagai atribut pembeda untuk setiap kasus, atribut ini bersifat unik dan nilainya bertipe numerik. Atribut galat (*error*), berisi pesan kesalahan yang diambil dari *compiler*. Atribut prog_salah dan prog_benar berisi contoh program yang salah dan program yang benar, dan atribut terakhir adalah solusi yang berisi pemecahan masalah dari pesan kesalahan beserta penjelasannya yang berbahasa Indonesia.

2.4 Case-Based Reasoning

Case-Based Reasoning (CBR) terdiri dari 4 siklus dalam pemecahan masalah: *retrieve*, *reuse*, *revise*, dan *retain*. Penelitian ini hanya fokus pada dua siklus yaitu *retrieve*, merupakan proses untuk mencari kesamaan antara kasus yang baru dengan kasus-kasus lama dan *reuse*, proses untuk memilih solusi yang tepat bagi pengguna. Tahap *retrieve* merupakan tahap yang sangat berpengaruh terhadap hasil atau solusi dari CBR karena pada tahap ini hasil kemiripan kasus ditentukan. Oleh karena itu, analisis terhadap fungsi similaritas perlu dilakukan dan disesuaikan dengan domain permasalahan.

Masukan yang diberikan pengguna ke sistem yang dihasilkan oleh *compiler* berupa kalimat bebas atau tidak terstruktur. *Natural Language Processing* (NLP) digunakan sebagai pra-proses untuk CBR untuk mengubah fitur yang tidak terstruktur menjadi terstruktur dan sesuai dengan format masukan untuk sistem CBR. Pengguna memberikan masukan/masalah menggunakan bahasa alami berupa kalimat. Kalimat tersebut diproses oleh NLP setelah itu baru diproses oleh CBR. Hasil pemrosesan dari CBR diberikan langsung ke pengguna.

Parsing merupakan salah satu proses dalam NLP yang digunakan untuk menganalisis sintak dan semantik dari suatu kalimat. Dalam penelitian ini, proses parsing digunakan untuk pemecahan kalimat menjadi kata-kata, setiap kata-kata yang dihasilkan dibuat bobot kata yang selanjutnya digunakan dalam fungsi similaritas CBR. Pembobotan dibuat guna untuk membedakan kata-kata spesifik dengan kata-kata umum. Kata-kata spesifik mempunyai bobot yang lebih besar dibandingkan dengan kata-kata

umum karena itulah digunakan metode pemberian bobot untuk setiap kata yang ada pada *target-case*. Pada Persamaan (1 sampai 8) merupakan rumusan untuk mencari bobot setiap kata *target-case* yang dihasilkan dari *query source-case*.

$S \rightarrow$ kelas *source-case*

$T \rightarrow$ kelas *target-case*

$$T = t_i = \{t_1, t_2, t_3 \dots t_n\} \quad (1)$$

$$J = j_i = \{j_1, j_2, j_3 \dots j_n\} \quad (2)$$

$$j_i = \sum_i^n t_i; \quad t_i \subset (Q = \text{Query } S) \quad (3)$$

$$n_j = \sum_i^n j_i \quad (4)$$

$$J' = j'_i = \frac{j_i}{n_j} \quad (5)$$

$$\text{maks} = \max(J') \quad (6)$$

$$\text{min} = \min(J') \quad (7)$$

$$W = w_i = (\text{maks} - j'_i + \text{min}) \quad (8)$$

Keterangan:

- S : *source-cases*
- T : *target-case*
- J : jumlah kata untuk setiap kata dalam kelas target
- n_j : jumlah total J
- J' : normalisasi dari J
- maks : nilai maks dari J'
- min : nilai min dari J'
- W : bobot-kata

CBR menggunakan fungsi similaritas untuk menghitung kesamaan antara kasus baru dengan kasus lama atau kasus yang berada dalam basisdata kasus (*case-base*). Fungsi similaritas yang digunakan dalam penelitian ini dapat dilihat pada Persamaan (9, 10 dan 11).

$$\text{sim}(S_i, T) = \frac{\sum_i^n f(S, T_i)}{f_p(S, T) + \sum_i^n w_i} \quad (9)$$

dimana,

$$f(S, T_i) = \begin{cases} 1 + w_i; & T_i \subset S \\ 0; & T_i \notin S \end{cases} \quad (10)$$

dan

$$f_p(S, T) = \begin{cases} \sum_i^n S_i; \sum_i^n S_i \geq \sum_i^n T \\ \sum_i^n T_i; \sum_i^n S_i < \sum_i^n T \end{cases} \quad (11)$$

Keterangan:

S : *source-cases*;

T : *target-case*;

w : bobot-kata;

$f(S, T_i)$: fungsi untuk menghitung kemiripan;

$fp(S, T)$: fungsi untuk mencari banyak kata S dan T .

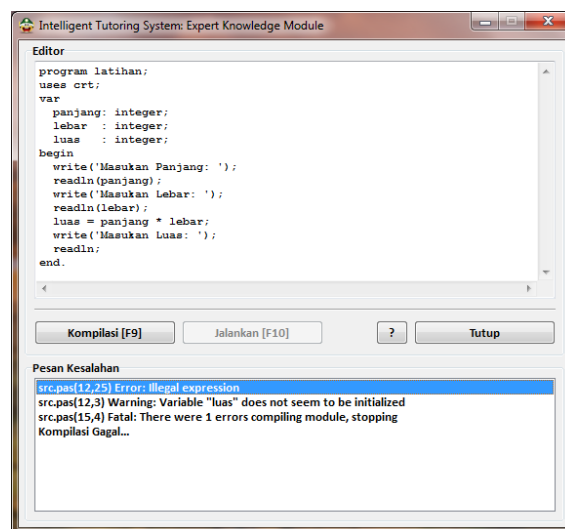
Berdasarkan perhitungan dari fungsi similaritas maka didapat rangking kasus. *Source-case* dengan rangking yang terbaik dijadikan solusi bagi *target-case*.

3. Hasil dan Analisis

Pengujian dilakukan pada penelitian ini bersifat pengujian kualitatif. Dimana tingkat kebenaran solusi yang diberikan oleh sistem tergantung dari subjektivitas pengguna. Mekanisme pengujian dilakukan dengan cara melakukan percobaan sebanyak beberapa kali dan pengujian tersebut dihitung persentase ketepatan solusi yang diberikan oleh sistem. Dari penelitian yang dilakukan dapat diperoleh 2 hasil penelitian yaitu perangkat lunak dan hasil pengujian. Perangkat lunak merupakan aplikasi yang mengimplementasikan arsitektur sistem yang dapat mengolah *input* dan *output*, serta digunakan sebagai alat uji untuk metode-metode yang digunakan. Pengujian merupakan cara yang digunakan untuk mengevaluasi metode dan mencari tingkat akurasi kebenaran dari solusi yang dihasilkan oleh sistem.

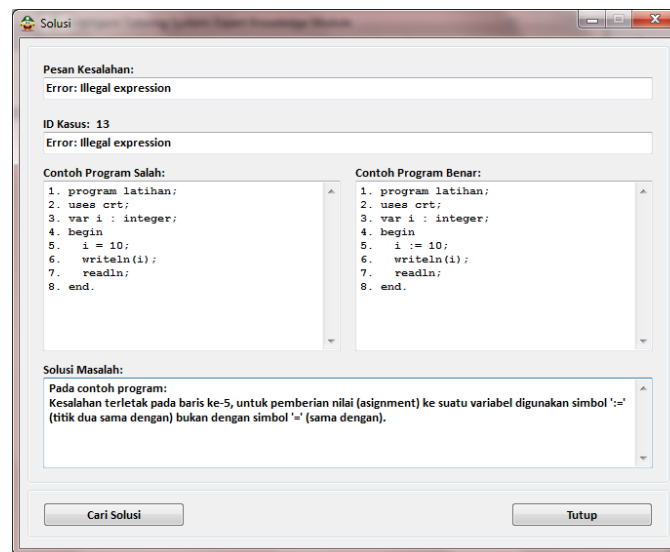
3.1 Perangkat Lunak Uji

Penelitian ini menghasilkan dua antarmuka yaitu antarmuka utama dan antarmuka solusi yang dapat dilihat pada Gambar 2 dan Gambar 3.



Gambar 2: Antarmuka Utama

Antarmuka pada Gambar 2 memiliki 4 buah komponen yaitu: 1) Editor, area tempat menuliskan kode program; 2) tombol [Kompilasi] digunakan untuk proses kompilasi kode program; 3) tombol [Jalankan] digunakan untuk menjalankan program yang berhasil dikompilasi; 4) Pesan Kesalahan, area tempat menampilkan pesan kesalahan yang terjadi. Antarmuka Gambar 3 ditampilkan ketika pengguna menekan pesan kesalahan pada Gambar 2. Terdapat 7 komponen: 1) Pesan Kesalahan, berisi pesan yang diambil dari gambar 6; 2) ID Kasus, berisi nomor dan kasus yang sama dengan pesan kesalahan; 3) Contoh Program Salah, berisi contoh program yang salah; 4) Contoh Program Benar, berisi contoh program yang benar; 5) Solusi Masalah, berisi solusi dari pesan kesalahan yang dicari; 6) Tombol Cari Solusi, tombol yang digunakan untuk memulai proses pencarian solusi.



Gambar 3: Antarmuka Solusi

3.2 Hasil Percobaan

Pengujian dilakukan dengan cara mencoba kode program secara berulang-ulang dan dicari kesalahan yang terjadi. Pada Tabel 1 dapat dilihat contoh pengujian kode program seperti pada contoh nomor 1 untuk pengujian simbol (;) titik koma.

Tabel 1: Hasil Percobaan

| No. | Kesalahan | Solusi |
|-----|----------------------------------|----------|
| 1 | Kurang titik koma (;) | BERHASIL |
| 2 | Salah ketik keyword | BERHASIL |
| 3 | Unit tidak dikenali | BERHASIL |
| 4 | Simbol pembuka string | GAGAL |
| 5 | Simbol penutup string | BERHASIL |
| 6 | Simbol pembuka kurung | GAGAL |
| 7 | Simbol penutup kurung | BERHASIL |
| 8 | Keyword write atau writeln | BERHASIL |
| 9 | Keyword read atau readln | GAGAL |
| 10 | Kesalahan simbol pemberian nilai | BERHASIL |

Penelitian ini belum mendapatkan akurasi yang pasti dari solusi yang diberikan oleh sistem. Hal ini terjadi karena pengujian hanya dilakukan secara subjektif oleh peneliti sendiri. Hipotesis sementara yang didapat untuk meningkatkan akurasi adalah menambahkan sebanyak mungkin data kasus sehingga lebih banyak kasus yang dapat dikenali oleh sistem.

4. Kesimpulan

Sistem *Expert-Knowledge Module* telah dikembangkan untuk *Intelligent Tutoring System* berbasis *Case-Based Reasoning* dalam ranah domain pemrograman komputer. Berdasarkan pengujian dapat dilihat bahwa terdapat beberapa kekurangan dari sistem yang dibangun diantaranya adalah sedikitnya pesan kesalahan yang dapat dideteksi oleh sistem hal ini dikarenakan sedikit kasus yang tersimpan dalam basisdata kasus. Hasil Solusi untuk beberapa kasus yang dikenali seperti contoh pada Gambar 2 dan 3 cukup memberikan keterangan yang signifikan dan penjelasan yang langsung tepat sasaran. Pengujian masih bersifat subjektif, sehingga penelitian ini belum dapat memberikan tingkat akurasi secara kuantitatif. Ada beberapa saran yang dapat digunakan untuk perbaikan dan penelitian berikutnya diantaranya adalah: 1) Perbanyak kasus sehingga setiap kesalahan kode program yang ditemui dapat dideteksi oleh sistem; 2) Lakukan pengujian kualitatif secara umum dengan mengadakan *survey* sehingga dapat memberikan akurasi kualitatif berdasarkan sudut pandang pengguna, dan 3) Lengkapi sistem untuk siklus *revise* dan *retain* pada CBR.

Referensi

- [1] H. S. Nwana, "Intelligent tutoring systems: An overview. *Artificial Intelligence Review*", 4, 251-277, 1990.
- [2] R. Nkambou, R. Mizoguchi, and J. Bourdeau, "Advances in intelligent tutoring systems", Heidelberg: Springer, 2010.
- [3] V. K. Aleven, J. Mostow, *Journal International Conference on Intelligent Tutoring System 10th June 14 – 18 2010. Pittsburgh, PA, USA, 2010.*
- [4] J. R. Carbonell, "AI in CAI: an artificial intelligence approach to computer-assisted instruction", *IEEE Transactions on Man-Machine Systems*, II, 190-202, 1970.
- [5] W. Swoboda, F.M. Zwiebel, R. Spitz, and L. Gierl, "A case-based consultation system for postoperative management of liver-transplanted patients", *Proceedings of the 12th MIE Lisbon*, IOS Press, Amsterdam, pp. 191-195, 1994.
- [6] I. Watson, "Applying Case-Based Reasoning: Techniques for Enterprise Systems", Morgan Kaufmann Publisher Inc. San Franscisco: California, 1997.