

# PERANCANGAN PENGEMBANGAN PROTOTYPE *INTELLIGENT TUTORING SYSTEM* UNTUK PEMBELAJARAN *IDIOMATIC EXPRESSIONS*

Yustina Niken Damayanti<sup>(1)</sup>

## Abstrak:

Almost students are lacking the necessary skills to solve translation problems. This paper is concerned with the development of an Intelligent Tutoring System (ITS) for teaching english translation.

We investigate the issues of how to generate useful feedbacks and explanations to students in an ITS focused on learning idiomatic expressions translation. In this system, students attempt to study their own questions of the idiomatic expressions problem. The system then generates the explanations in details of how to solve that problem using defined translation rules, and to avoid erroneous problem solving.

In addition, when the students attempt to solve the idiomatic expressions problems, the system compares the student answer with the system answer and then generates the explanations using a problem-solving tree which is generated for a particular problem. The system syntactically compares the two solutions and points out discrepancies between them when the student's solution is incorrect.

**Kata Kunci :** *Intelligent Tutoring System (ITS), idiomatic expressions, translation*

## 1. Pendahuluan

Bahasa Inggris dapat dikatakan sebagai salah satu pelajaran yang sulit, baik dalam pengajaran maupun pembelajarannya. Selain itu, bahasa Inggris membutuhkan kerja keras dan jumlah latihan yang banyak. Dalam pembelajaran bahasa Inggris sebenarnya meliputi pemahaman jenis kata dan konteks yang harus dipelajari secara bersamaan. Akan tetapi, hal ini tidak dapat tercapai sepenuhnya, karena ada beberapa faktor penghalang, seperti terbatasnya sumber daya.

Untuk mengatasi hal ini sebenarnya sudah cukup banyak alat bantu komputer yang dikembangkan untuk membantu proses pembelajaran tersebut, tetapi alat bantu ini masih memiliki kekurangan, yaitu minimnya interaksi *user* dengan sistem. Sebagai penyelesaiannya, saat ini telah dikembangkan sebuah pendekatan sistem baru yang disebut dengan *intelligent tutoring system*, dimana sistem ini mampu untuk menghasilkan penjelasan yang berguna dan *feedback*, sehingga sistem dapat memberikan informasi untuk memperbaiki cara belajar dan pemahaman kepada *user*.

Domain permasalahan dalam sistem yang dicoba untuk dikembangkan adalah permasalahan dalam menterjemahkan *idiomatic expressions* (ekspresi idiom), meskipun sistem yang dikembangkan ini telah dianggap mampu untuk menghasilkan penjelasan penyelesaian masalah dasar-dasar penterjemahan. Sistem ini nantinya dirancang untuk mengecek jawaban *user* dalam menterjemahkan ekspresi idiom, dan dapat memberikan penjelasan tentang kesalahan yang telah dilakukan *user* dalam menterjemahkan.

## 2. Landasan Teori

*Intelligent tutoring system* untuk pembelajaran ekspresi idiom dirancang untuk :

memfasilitasi *user* dalam pembelajaran bahasa Inggris.

mendiagnosa tingkat kesalahan *user* dan memberikan penjelasan mengenai kesalahannya secara efektif.

menawarkan *user* sebuah sistem monitoring yang berisi progres hasil belajar secara relevan menggunakan data statistik.

<sup>(1)</sup> Yustina Niken Damayanti, Mahasiswa Magister Ilmu Komputer Universitas Gajah Mada, Gedung SIC Lantai III, FMIPA-UGM, Sekip Utara, Bulaksumur, Yogyakarta ta. Email : yustina\_niken@yahoo.com

Komputer telah banyak digunakan dalam bidang pendidikan dengan berbagai cara, dan tak kalah banyaknya dengan solusi yang telah diberikan. Sistem edukasi dengan menggunakan komputer pertama kali disebut dengan *Computer-Based Training (CBT)* dan *Computer Aided Instruction (CAI)*. Kedua sistem ini ternyata hanya bersifat instruksional dan hanya bersifat searah saja, sehingga *user* kurang berinteraksi dengan sistem. Karena keterbatasan inilah kemudian dikembangkan *intelligent tutoring system*.

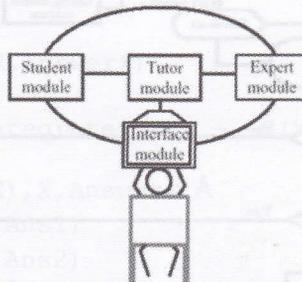
*Intelligent tutoring system* ini diharapkan mampu untuk (i) berinteraksi dengan *user* untuk menentukan level pengetahuan *user*, (ii) memiliki bank pengetahuan yang besar dan komprehensif, (iii) mengadaptasi strategi instruksional terhadap performa *user*. Meskipun tidak ada bentuk arsitektur standar untuk *intelligent tutoring system*, akan tetapi *intelligent tutoring system* ini secara arsitektur konseptual memiliki empat komponen pembangun utama, yaitu :

*Expert Module* : terdiri atas informasi-informasi mengenai domain pengetahuan, berupa fakta-fakta dan konsep-konsep yang sering dikemukakan. *Expert module* ini juga memberikan sumber pengetahuan yang dibutuhkan untuk pemecahan masalah dan penjelasan tentang konsep dan respon terhadap *user*. Di dalam *expert module* ini mempunyai sub komponen *generating process* dan *pre-processing*.

*Student Module* : yang juga termasuk preferensi *user*. Sejarah *user tutoring* digunakan untuk meningkatkan performa sistem *tutoring*.

*Tutor Module* : terdiri atas *planner* yang menggunakan aturan. *Planner* ini berfungsi untuk menentukan jenis *tutoring* untuk setiap pokok bahasan yang cocok. *Tutor module* ini nantinya akan memonitor *user's action* dan mengadaptasinya sebagai bentuk respon terhadap *user*. Sebenarnya baik di dalam *student module* maupun *tutor module* memiliki sub komponen yang sama untuk *explaining process*, *error diagnosis process*, dan *matching process* (untuk menentukan *incorrect answer* dan *correct answer*).

*Interface Module* : berfungsi untuk mengatur aliran komunikasi antara *user* dengan *intelligent tutoring system*.



Gambar 1. Gambaran Umum Arsitektur Konseptual *Intelligent Tutoring System*

### 3. Analisis dan Perancangan

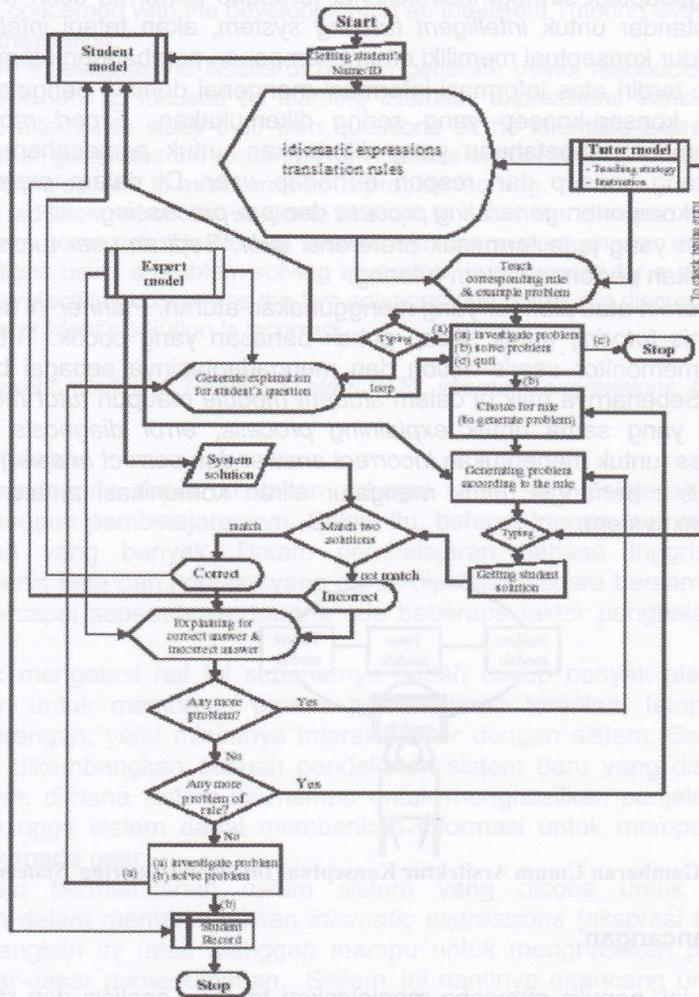
#### 3.1 Analisis Masalah

Dalam tahapan ini, penulis mencoba menjelaskan tentang analisis dan rancangan alur dari sistem.

Salah satu analisis yang dilakukan adalah analisis pengguna sistem atau *user*. Adapun *user* atau pengguna sistem ini adalah mahasiswa (*student*).

Pertama-tama pada saat memulai sesi pembelajaran *user* memasukkan identitasnya, berupa nama dan NIM yang kemudian akan disimpan dalam *student record*. Tahapan berikutnya *user* akan dibimbing dalam aturan-aturan dan teknik-teknik translasi untuk ekspresi idiom dan menyelesaikan beberapa contoh permasalahan yang terkait. Setelah *user* selesai melihat presentasi singkat mengenai aturan dan teknik translasi untuk ekspresi idiom, sistem akan memiliki tiga tujuan pilihan untuk memproses penyelesaian masalah *user*, yaitu (a) investigasi terhadap masalah yang ada, (b) penyelesaian masalah, dan (c) keluar dari sistem. Berdasarkan ketiga pilihan ini, diasumsikan bahwa *user* akan mengecek beberapa contoh permasalahan. Ketika *user* menginputkan pertanyaan seputar aturan dan teknik translasi untuk ekspresi idiom berupa informasi tentang kesalahan yang mungkin terjadi (jika ada) kepada sistem.

Kemudian sistem meresponnya dengan cara mengevaluasi jawaban *user* dengan mencocokkan jawabannya dengan jawaban sistem. Jika cocok, maka sistem akan menampilkan informasi bahwa jawaban *user* adalah benar. Akan tetapi jika tidak cocok, maka sistem akan memberikan penjelasan dan menampilkan jenis kesalahan yang terjadi serta jawaban *user* dan jawaban sistem sebagai pembandingan. Sebagai tambahan dalam sistem ini semua jawaban *user* disimpan di dalam *student record stores*, sehingga *user* dapat mem-browse semua jawabannya, baik yang benar maupun salah beserta penyelesaiannya, dan sistem ini dapat menyimpan *user progress record*, selama *user* melakukan tutoring dengan sistem ini.



Gambar 2. Flowchart *Intelligent Tutoring System*

### 3.2 Rancangan Sistem

#### 3.2.1 Interface Model

Untuk *interface model*, sistem ini akan menggunakan *Graphic User Interface (GUI)* yang diimplentasikan dengan komponen-komponen *intelligent* berbasis HTML. Komponen ini memproses output screen yang dapat di-cover dalam Prolog (*Programming in Logic*), untuk menghasilkan *educational environment* yang baik.

HTML berbasis GUI berfungsi untuk output yang bertujuan untuk pengajaran, pembelajaran, penjelasan, dan instruksi aturan dan teknik translasi ekspresi idiom.

Prolog berbasis GUI berfungsi untuk mengisikan input yang bertujuan untuk mengecek beberapa permasalahan dalam translasi ekspresi idiom, dan melakukan penyelesaian terhadap permasalahan dan kesalahan *user*.

#### 3.2.2 Student Model

*Student model* ini akan menyimpan *user history* yang berisi *student record* dan *student updates*, termasuk nama, NIM, jumlah permasalahan yang telah terkoreksi, dan jumlah jawaban yang benar dan salah. Ketika *user* melakukan *tutoring* dan pengecekan permasalahan, maka sistem akan secara otomatis melakukan *update*. Sehingga pada akhir penggunaan sistem dapat ditampilkan progres aktifitas *user*.

### 3.2.3 Tutor Model

*Tutor model* berfungsi memberikan pengajaran dan penjelasan mengenai aturan translasi ekspresi idiom beserta koreksi kesalahan penulisan, seperti menampilkan sebuah permasalahan beserta penjelasan metode penyelesaiannya. Contoh :

*User* salah mengetikkan kata :

*Baking up the wrong tree*

Maka akan diralat menjadi :

*Barking up the wrong tree*

### 3.2.4 Expert Model

*Expert model* tersusun atas domain pengetahuan yang akan diajarkan. Komponen *intelligent tutoring system* ini berfungsi dalam menentukan tugas untuk *generate* permasalahan-permasalahan dan mengevaluasi kekurangan untuk solusi *user*, serta menentukan penjelasan yang dinilai tepat berdasarkan solusi *user*.

### 3.2.5 Generating Process

*Generating process* ini akan mencocokkan atau membandingkan jawaban atau penyelesaian *user* dengan penyelesaian sistem. Untuk mendapatkan hasil perbandingan ini, dapat digunakan *pseudocode* menggunakan prolog, seperti berikut ini :

```
question(+Term, -Answer) A
    (Term=F(X) + G(X)
    integrate(F(X) + G
    (X), X, Answer) )
    (Term=F(X)
    integrate(F(X) , X, Answer) )
    (Term=F(X) - G(X)
    integrate(F(X) - G(X), X, Answer) ) .
```

```
integrate(F(X) + G(X), X, Answer) A
    integrate(F(X), X, Ans1)
    integrate(G(X), X, Ans2)
    Answer=Ans1 + Ans2.
```

```
integrate(F(X) - G(X), X, Answer) A
    integrate(F(X), X, Ans1)
    integrate(G(X), X, Ans2)
    Answer=Ans1 - Ans2.
```

```
integrate(F(X), X, Answer) A
    F(X) = K
    Answer=K*X.
```

### 3.2.6 Pre-processing

*Pre-processing* berfungsi berfungsi untuk mengkonversi input menjadi bentuk normal sistem.

### 3.2.7 Matching Process

Sedangkan *matching processing* berfungsi untuk mencocokkan antara jawaban penyelesaian *user* dengan sistem, sehingga akan menampilkan jumlah jawaban yang salah dan yang benar.

### 3.2.8 Error Diagnosis Process

Bertujuan untuk menampilkan solusi yang diinputkan oleh *user* dengan solusi yang dihasilkan oleh sistem. Kedua solusi ini ditampilkan, sehingga *user* dapat mengetahui letak kesalahannya (jika jawaban *user* salah).

**3.2.9 Explaining Process**

Bertujuan untuk menjelaskan letak kesalahan *user* dalam mencari solusi permasalahan.

Contoh :

<b>Idiomatic Translation Case</b>		
<i>Putting the cart before the horse</i>		
<b>System Solution</b>	: doing things in the wrong order	
<b>Your Solution</b>	: doing something dangerous	
<b>Criteria</b>	<b>System</b>	<b>User</b>
<b>Sentence</b>	Æ things in the wrong order	something dangerous
<b>Match</b>	Æ doing	= doing

**4. Kesimpulan**

Dalam tulisan ini, dapat disimpulkan bahwa rancangan *intelligent tutoring system* untuk translasi atau penterjemah ekspresi idiom (*idiomatic expressions*) secara umum terfokus pada proses penterjemahan ekspresi idiom, dan dirancang untuk membantu *user* dalam menyelesaikan permasalahan translasi dengan menggunakan tampilan penjelasan, sehingga memudahkan *user* untuk berinteraksi dengan sistem dalam memahami proses translasi ekspresi idiom.

**5. Daftar Pustaka**

Bratko, Ivan. (2001) . *Prolog Programming for Artificial Intelligence 3<sup>rd</sup> Edition*. Addison-Wesley  
 Firebaugh, Morris W., *Artificial Intelligent A Knowledge-Based Approach*, PWS-KENT Publishing  
 Company, 1998.  
 Gazdar, Gerald & Chris Mellish. *Natural Language Processing in Prolog : An Introduction to  
 Computational Linguistic*. Mackays of Chatham plc, Chatham, Kent : Great Britain. 1989.  
 Sri Hartati, & Bernard Renaldy Suteja . 2008 . *Implementasi Intelligent Tutoring System dalam  
 Pengembangan Sistem E-learning*. Prosiding KNSI 2008 : Yogyakarta

**6. Biografi Penulis**

**Yustina Niken Damayanti**

Menyelesaikan program S1 jurusan Ilmu Komputer Universitas Sanata Dharma Yogyakarta pada tahun 2005. Saat ini sedang menyelesaikan program S2 di jurusan Ilmu Komputer Universitas Gajah Mada Yogyakarta dengan minat ilmu pada sistem cerdas dan komputasi. Informasi lebih lanjut tentang penulis ini dapat melalui alamat e-mail : [yustina\\_niken@yahoo.com](mailto:yustina_niken@yahoo.com)