

GAME LABIRIN BERBASIS PENGOLAH BAHASA ALAMI UNTUK PENGENALAN LOGIKA PEMROGRAMAN SEDERHANA

Chandra Kusuma Dewa

Jurusan Teknik Informatika, Fakultas Teknologi Industri
Universitas Islam Indonesia, Yogyakarta

ckusumadewa@gmail.com

Abstrak

Untuk mengenalkan konsep pemrograman kepada siswa sekolah menengah, sebuah game pembelajaran pemrograman, CodeCombat telah dikembangkan. Dengan CodeCombat, siswa dikenalkan dengan konsep pemrograman melalui petualangan karakter dalam game yang dapat dikendalikan dengan mengetikkan sejumlah perintah tertentu. Meskipun demikian, CodeCombat harus dimainkan secara online serta masih menggunakan instruksi dalam Bahasa Inggris. Pada makalah ini, diusulkan rancangan game labirin berbasis pengolah bahasa alami yang diturunkan dari CodeCombat. Siswa nantinya dapat mengetikkan perintah-perintah dalam bahasa pemrograman sederhana yang ada dalam game untuk mengarahkan karakter agar dapat keluar dari labirin. Hasil dari penelitian ini berupa sebuah game offline pembelajaran logika pemrograman sederhana yang dilengkapi dengan modul pengolah bahasa alami untuk perintah arah dan perintah gerak dalam Bahasa Indonesia.

Kata Kunci: Pengenalan Logika Pemrograman; Game Labirin; Pengolah Bahasa Alami.

1. Pendahuluan

Terdapat banyak aplikasi komputer yang dapat digunakan untuk mengenalkan konsep pemrograman kepada siswa atau pelajar. Dari beberapa aplikasi tersebut, terdapat dua aplikasi yang cukup populer, diantaranya adalah Logo (Papert, 1980) yang menggunakan pendekatan berbasis teks, serta Scratch (Maloney, dkk., 2008) yang menggunakan pendekatan berbasis visual ataupun multimedia. Dengan menggunakan Logo, siswa dapat mempelajari konsep pemrograman dengan cara mengetikkan beberapa baris perintah sederhana untuk menggambar suatu objek tertentu. Sementara, dengan menggunakan Scratch, siswa dapat mempelajari konsep pemrograman melalui pembuatan aplikasi multimedia sederhana, seperti animasi karakter tertentu, dengan cara menyusun kombinasi dari beberapa blok perintah yang sudah disediakan dalam aplikasi.

Selain kedua aplikasi tersebut, terdapat pula aplikasi pembelajaran pemrograman yang mencoba menggunakan pendekatan yang sedikit berbeda, yakni pendekatan berbasis permainan. Hal tersebut cukup beralasan mengingat penggunaan *game* yang diintegrasikan dengan kegiatan belajar mengajar dapat

meningkatkan minat dan motivasi belajar siswa (Ahdell & Andresen, 2001). CodeCombat (<http://codecombat.com>), sebuah game online pembelajaran pemrograman, mencoba menggabungkan kedua pendekatan yang dimiliki oleh Logo dan Scratch. Pada game tersebut, pemain akan diajak berpetualang dengan mengetikkan sekumpulan perintah untuk dapat memainkan permainan pada setiap *level*-nya. Untuk mengenalkan konsep pemrograman, perintah-perintah tersebut selanjutnya diwujudkan ke dalam bentuk-bentuk tertentu yang menyerupai pemanggilan fungsi pada bahasa JavaScript.

Karena CodeCombat adalah *game online*, maka tentu saja pemain harus selalu terhubung dengan Internet serta harus memiliki koneksi Internet yang cukup cepat dan stabil agar aplikasi ini dapat berjalan dengan baik. Oleh karena itu, diperlukan aplikasi sejenis yang sekiranya dapat diterapkan untuk wilayah-wilayah dengan infrastruktur Internet yang kurang begitu baik. Selain itu, instruksi ataupun bahasa pemrograman dalam CodeCombat untuk saat ini masih dalam bentuk Bahasa Inggris. Akan jauh lebih baik jika bahasa pemrograman yang ada dalam *game* juga dikembangkan untuk bahasa-bahasa selain Bahasa Inggris, seperti Bahasa Indonesia.

Pada penelitian ini, dikembangkan sebuah *game offline* yang bertujuan untuk mengenalkan logika dan konsep pemrograman sederhana yang diturunkan dari CodeCombat. *Game* selanjutnya diwujudkan dalam bentuk permainan labirin, dimana pemain ditugaskan untuk mengarahkan karakter agar dapat keluar dari labirin dengan menggunakan bahasa pemrograman sederhana yang perintahnya merupakan perintah arah dan perintah gerak dalam Bahasa Indonesia. *Game* juga dilengkapi dengan sebuah modul pengolahan bahasa alami untuk melakukan proses pengecekan terhadap kesalahan sintaks, serta menerjemahkan perintah-perintah tersebut untuk dapat menggerakkan karakter.

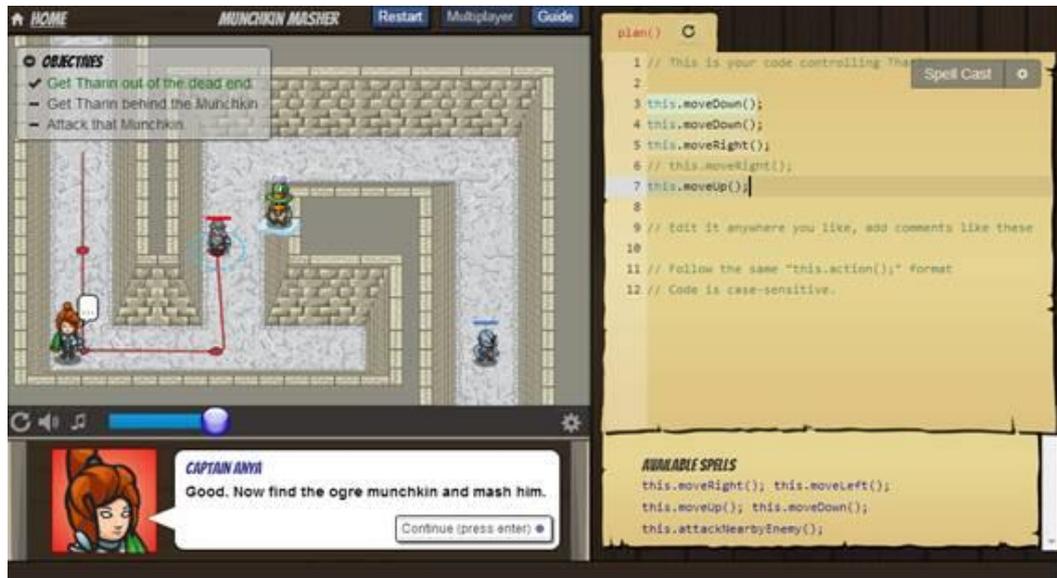
Sistematika penulisan yang digunakan dalam makalah ini adalah sebagai berikut: pada bagian 2 akan dibahas mengenai tinjauan pustaka yang digunakan dalam penelitian ini, pada bagian 3 akan dibahas mengenai perancangan sistem, pada bagian 4 akan dibahas mengenai hasil dan pengujian sistem, serta pada bagian 5 akan dibahas mengenai mengenai kesimpulan dan saran.

2. Tinjauan Pustaka

2.1 *Game* CodeCombat

Game CodeCombat terdiri dari beberapa *level* permainan. Untuk setiap *level*-nya, pemain akan diberikan sebuah misi tertentu yang dapat diselesaikan

dengan cara menggerakkan karakter melalui pengetikan sejumlah rangkaian perintah tertentu sebagaimana ditunjukkan pada Gambar 1. CodeCombat mencoba menggunakan keunggulan multimedia yang terinspirasi dari Scratch. Untuk dapat menarik minat siswa, proses pembelajaran logika dan pemrograman selanjutnya dibungkus dalam sebuah permainan.



Gambar 1 Salah Satu Level dalam CodeCombat

Selain pendekatan multimedia yang terinspirasi dari Scratch, CodeCombat juga menggunakan pendekatan tekstual yang terinspirasi dari aplikasi Logo. Logika siswa selanjutnya akan terbangun melalui proses penyusunan baris-baris perintah yang diketikkan untuk mencapai suatu tujuan tertentu. Siswa juga secara tidak langsung dapat belajar tentang sintaks pemrograman serta konsep kesalahan sintaks dalam pemrograman karena aplikasi akan mengeluarkan pesan-pesan kesalahan jika perintah yang diketikkan salah secara sintaksis.

2.2 Pengolahan Bahasa Alami

Menurut Suciadi (2001), pengolahan bahasa alami adalah salah satu cabang dari ilmu kecerdasan buatan yang mempelajari tentang komunikasi antara manusia dengan komputer melalui bahasa alami. Lebih jauh lagi, pengolahan bahasa alami dikembangkan agar komputer dapat mengerti dan memahami bahasa alami yang diberikan serta mampu memberikan respon hasil pengolahan sesuai yang diinginkan (Indra & Hartati, 2014). Pengolahan bahasa alami bertujuan untuk memahami perintah bahasa alami serta melakukan respon yang sesuai terhadap perintah tersebut.

Pengolahan bahasa alami sendiri terdiri dari tiga bagian, yakni *parser*, sistem representasi pengetahuan serta pengolahan *output* (Soyusiawaty, 2008). *Parser* akan mengambil kalimat input bahasa alami dan kemudian menguraikannya ke dalam bentuk gramatikal. Sistem representasi pengetahuan akan menganalisis *output parser* dan menganalisis hasilnya sesuai aturan produksi (*grammar*), serta *output translator* akan merepresentasikan langkah-langkah yang sesuai.

Dalam *game* pada usulan penelitian ini, pengolahan bahasa alami akan diwujudkan menjadi sebuah modul tersendiri. Perintah-perintah yang dituliskan oleh siswa selanjutnya akan diolah menggunakan modul tersebut yang menggunakan *tools* ANTLR (Parr & Quong, 1995) untuk melakukan proses validasi dan proses penterjemahan. Jika perintah-perintah yang diketikkan siswa bebas dari kesalahan sintaksis maka karakter akan dapat digerakkan sesuai urutan perintah hingga dapat keluar dari labirin.

Tabel 1 Rancangan materi logika dan pemrograman

Materi	Implementasi
Pengenalan posisi/letak karakter utama	Diwujudkan dengan merancang papan menjadi bentuk <i>grid-grid</i>
Pengenalan struktur bahasa pemrograman sederhana	Diwujudkan menggunakan <i>grammar</i> pada bahasa sederhana yang dipakai dalam <i>game</i>
Pengenalan kesalahan sintaksis sederhana dalam pemrograman	Diwujudkan dalam modul pengolahan bahasa alami yang ada dalam <i>game</i>
Pengenalan logika sederhana	Diwujudkan dalam <i>gameplay</i> yang mengharuskan pemain mengetikkan sekumpulan instruksi untuk mencapai tujuan tertentu dengan memperhatikan rancangan papan yang berbentuk <i>grid-grid</i>

3. Rancangan *Game* Labirin

3.1 Rancangan Materi Logika dan Pemrograman

Detail rancangan materi logika dan pemrograman pada *game* labirin yang diusulkan selanjutnya ditunjukkan pada Tabel 1. Dari uraian materi pada tabel tersebut, selanjutnya dapat dilihat bahwa pengenalan logika sederhana diimplementasikan melalui tugas yang harus dilakukan oleh siswa, yakni bagaimana mengarahkan karakter agar dapat keluar dari labirin melalui urutan-urutan serta rangkaian-rangkaian perintah untuk mencapai suatu tujuan tertentu. Hal tersebut sesuai dengan konsep algoritma dalam pemrograman yakni urutan langkah-langkah untuk memecahkan suatu masalah tertentu (Munir, 2011). Selain itu, siswa juga dikenalkan bahwa dalam pemrograman, perintah-perintah yang dapat diketikkan memiliki tata bahasa serta sintaks tertentu yang harus diikuti.

3.2 Rancangan Antarmuka *Game*

Rancangan antarmuka *game* selanjutnya ditunjukkan pada gambar 2. Pada rancangan antarmuka tersebut, terlihat bahwa papan utama *game* dibagi menjadi *grid-grid* dengan koordinat tertentu yang membentuk suatu labirin. Pada papan utama, karakter serangga akan diletakkan dengan posisi acak yang ditandai dengan kotak berwarna merah, lalu siswa diminta untuk mengarahkan karakter serangga tersebut agar dapat keluar dari labirin melalui pintu yang posisinya juga acak, yang ditandai dengan kotak berwarna hijau.

Siswa dapat mengetikkan perintah-perintah arah dan gerak dengan sintaks dan *grammar* tertentu melalui *text area* yang terletak pada sisi sebelah kanan papan utama. Setelah urutan dan rangkaian perintah tersebut selesai diketikkan, siswa selanjutnya dapat menekan tombol untuk menjalankan karakter. Jika terdapat kesalahan sintaks ataupun *grammar*, pesan kesalahan selanjutnya akan ditampilkan pada teks area yang terletak di bawah papan utama.



Gambar 2 Rancangan papan utama game

3.3 *Grammar* Perintah Arah dan Gerak

Untuk dapat menggerakkan karakter pada *game* agar dapat keluar dari labirin, siswa dapat menggunakan ataupun mengkombinasikan perintah-perintah sederhana terkait arah dan gerak yang nantinya harus disusun berdasarkan urutan tertentu yang detailnya dapat dilihat pada Tabel 2.

Urutan perintah-perintah yang diketikkan oleh siswa nantinya harus mengikuti *grammar* yang dikembangkan dalam *game*. *Grammar* pada *game* berfungsi untuk mengenali perintah-perintah terkait arah dan gerak tersebut. Detail *grammar* yang digunakan pada *game* labirin selanjutnya ditunjukkan pada Gambar 3.

Tabel 2 Daftar perintah arah dan gerak dalam *game* labirin

Jenis Perintah	Detail Perintah
Perintah Arah	balik kanan
	hadap kanan
	hadap kiri
	hadap serong kanan
	hadap serong kiri
Perintah Gerak	maju satu langkah
	maju dua langkah
	maju tiga langkah
	maju empat langkah
	maju lima langkah
	maju enam langkah
	maju tujuh langkah
	maju delapan langkah
	maju sembilan langkah
	maju sepuluh langkah

```

commandList
  : 'mulai' (command)+ 'selesai';

command
  : faceCommand | moveCommand;

faceCommand
  : 'hadap' direction | 'balik kanan';

direction
  : 'kanan' | 'kiri'
  | 'serong kanan' | 'serong kiri';

moveCommand
  : 'maju' jumlahLangkah 'langkah';

jumlahLangkah
  : 'satu' | 'dua' | 'tiga' | 'empat' | 'lima'
  | 'enam' | 'tujuh' | 'delapan' | 'sembilan' | 'sepuluh';

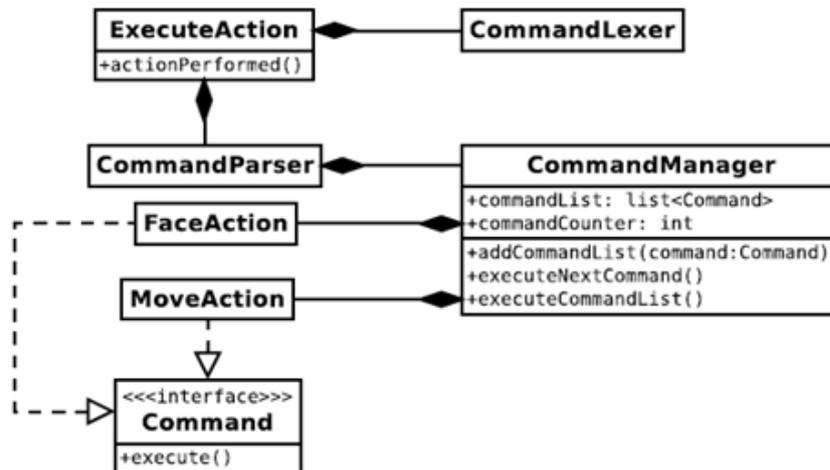
```

Gambar 3 *Grammar* pada *game* labirin

3.4 Diagram Kelas *Game* Labirin

Game labirin nantinya akan dikembangkan dengan menggunakan Bahasa Java berbasis Java Desktop (J2SE) serta menggunakan *library* ANTLR untuk memproses *grammar* dalam *game*. Dengan menggunakan *library* tersebut, *grammar* selanjutnya akan diolah sehingga dapat menghasilkan *parser* dan *lexer*

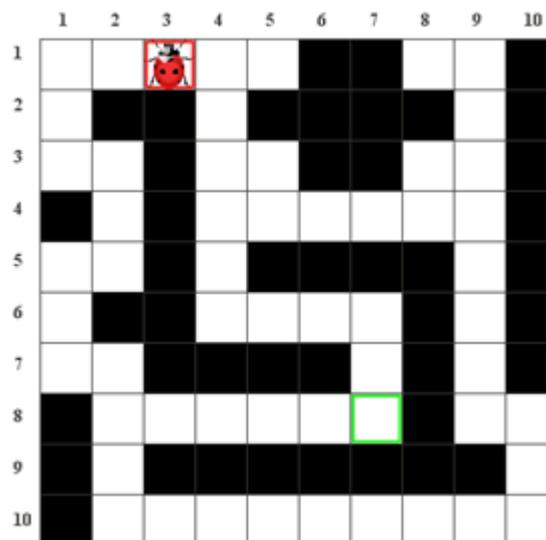
yang nantinya akan berperan menerjemahkan perintah berbasis bahasa alami yang diketikkan oleh siswa menjadi pemanggilan *method* pada objek-objek kelas Command serta kelas Action yang ada pada *game*. Detail diagram kelas untuk *game* labirin yang dikembangkan ditunjukkan pada Gambar 4.



Gambar 4 Diagram kelas *game* labirin

4. Hasil dan Pengujian

Proses pengujian selanjutnya dilakukan dengan cara mencoba *game* labirin yang telah berhasil dikembangkan. Pada tahapan pengujian, akan digunakan kasus papan utama *game* sebagaimana ditunjukkan pada Gambar 5. Proses pengujian akan dibagi menjadi dua *test cases*, yakni untuk urutan perintah yang sesuai *grammar* dalam *game* serta untuk urutan perintah yang tidak sesuai dengan *grammar* dalam *game*.



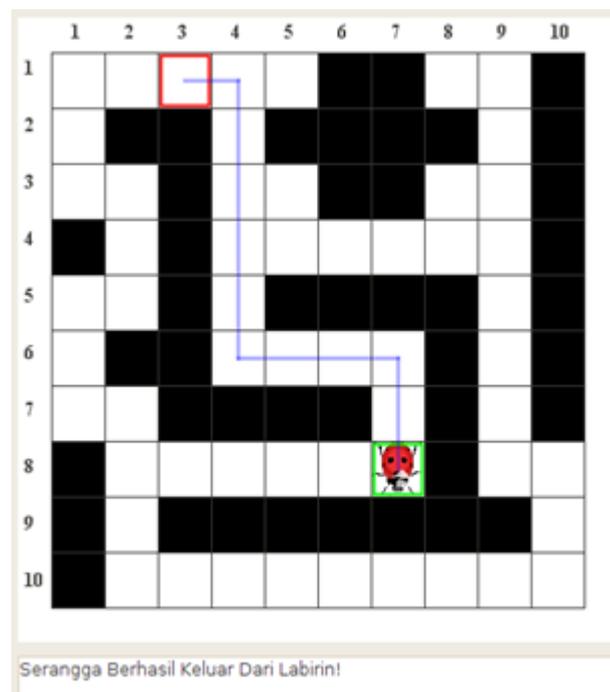
Gambar 5 Kasus papan utama *game* labirin

4.1 Pengujian Pertama

Pada pengujian yang pertama ini akan dilakukan pengetikan perintah yang sesuai dengan grammar yang ada pada *game* labirin. Detail urutan perintah yang digunakan selanjutnya ditunjukkan pada Gambar 6, sementara *output game* ditunjukkan pada Gambar 7. Dari gambar tersebut, terlihat bahwa aplikasi mampu mengolah bahasa alami menjadi perintah-perintah untuk menggerakkan karakter serangga untuk dapat keluar dari labirin.

```
mulai
hadap kanan
maju satu langkah
hadap kanan
maju lima langkah
hadap kiri
maju tiga langkah
hadap kanan
maju dua langkah
selesai
```

Gambar 6 Urutan perintah pengujian 1



Gambar 7 *Output game* labirin pengujian 1

4.2 Pengujian Kedua

Pada pengujian kedua, dilakukan pengetikan perintah yang tidak sesuai dengan *grammar* yang ada pada *game* labirin yang ditunjukkan pada Gambar 8. *Output game* untuk pengujian ini selanjutnya ditunjukkan pada Gambar 9. Dari *output* yang dihasilkan oleh *game* labirin, terlihat bahwa aplikasi dapat mengenali

kesalahan sintaksis dari perintah jika tidak perintah yang diketikkan tidak sesuai dengan *grammar* yang sudah ditentukan.

```
mulai
hadap kan
maju satu langkah
hadap kanan
maju 5 langkah
hadap kiri
maju tiga langkah
hadap kanan
maju dua langkah
selesai
```

Gambar 8 Urutan perintah pengujian 2

```
baris 2:9 kesalahan karakter '\n' seharusnya 'a'
baris 3:0 tidak ada alternatif untuk input 'maju'
baris 5:5 tidak ada alternatif untuk karakter '5'
baris 5:7 tidak ada alternatif untuk input 'langkah'
```

Gambar 9 Output game labirin pengujian 2

5. Kesimpulan

Dari pengujian yang telah dilakukan, maka dapat ditarik kesimpulan bahwa teknik pengolah bahasa alami telah berhasil diimplementasikan ke dalam *game* labirin yang dikembangkan. Selain itu, teknik pengolah bahasa alami yang diwujudkan ke dalam modul pada *game* labirin juga berhasil melakukan proses validasi serta proses penerjemahan perintah-perintah yang diketikkan.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Direktorat Penelitian dan Pengabdian Masyarakat Universitas Islam Indonesia (DPPM - UII) yang telah mendukung dan membiayai penelitian ini dengan Hibah Penelitian Dosen Pemula No. 079/Dir/DPPM/70/Pen.Pemula/XII/2014.

Daftar Pustaka

- Ahdell, R. & Andresen, G., 2001. Games and simulations in workspace eLearning: How to align eLearning content with learners needs. *Tesis*. Trondheim, Norway: Norwegian University of Science and Technology.
- Indra & Hartati, S., 2014. Aplikasi Pengolah Bahasa Alami untuk Info Gempa Bumi Terkini dengan Sumber Data pada Twitter @infobmkg. *Prosiding Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, pp. F7-F14.
- Maloney, J., Peppler, K., Kafai, Y.B., Resnick, M. & Rusk, N., 2008. Programming by Choice: Urban Youth Learning Programming with Scratch. *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, pp. 367-371.
- Munir, R., 2011. *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*. Bandung: Penerbit Informatika.
- Papert, S., 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.
- Parr, T. & Quong, R.W., 1995. ANTLR: A Predicated-ll(k) Parser Generator. *Software - Practice and Experience*, 25(7), pp. 789-810.
- Soyusiawaty, D., 2008. E-Translator with Rule Based Indonesia-Minang dan Minang Indonesia. *Jurnal Informatika*, 2(2), pp. 234-247.
- Suciadi, J., 2001. Studi Analisis Metode-Metode Parsing dan Interpretasi Semantik pada Natural Language Processing. *Jurnal Informatika*, 2(1), pp. 13-22.