

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Error Handling merupakan penanganan kesalahan pada berbagai macam keadaan dalam pemrograman. Dalam menulis *grammar* pada sistem sebelum diterjemahkan kedalam *source code*, sistem terlebih dahulu menganalisa kesalahan teks yang dimasukkan kedalam sistem. Menurut Ruslan Posevkin dan Igor Bessmertny pada penelitiannya, penanganan kesalahan pada natural language cukup sulit diterapkan dikarenakan diperlukan ketelitian dalam menyusun grammar dan sintaks serta percobaan yang tidak sedikit untuk mendapatkan hasil *error* yang diharapkan. Pada penelitian tersebut juga dikatakan bahwa penelitian ini cukup menarik dikarenakan peneliti harus mendapatkan solusi dengan hasil *error* dimana pada penelitian lainnya diharapkan dapat mendapatkan solusi untuk mencegah hasil error [1].

Beberapa penelitian sudah dilakukan untuk penerjemah bahasa alami ke mesin seperti translasi dari bahasa Russia ke struktur *data query* dalam bahasa C++ [1], dan penelitian lainnya dapat membuat proses penanganan masalah pada grammar dan teks [2][3]. Penelitian yang dilakukan oleh Torres, F., Hurtado, L. F., García, F., Sanchis, E., dan Segarra, E. dapat membuat penanganan masalah pada dialog stokastik [2]. Penelitian yang serupa juga dilakukan oleh Naghshnejad, M., Joshi, T., & Nair, V. N. dengan menggunakan proses *error handling* pada grammar menggunakan *deep learning* [3]. Akan tetapi, pada penelitian [2][3] tidak dapat menangani kesalahan dalam teks yang bersifat ambigu yang dapat menyebabkan *error* sehingga pengguna tidak dapat mengetahui kesalahan pada teks masukkan yang dapat memicu *error*. Penelitian yang dilakukan oleh Ruslan Posevkin dan Igor Bessmertny [1] memiliki fitur penanganan kesalahan untuk mengatasi teks masukkan yang *error*, tetapi sistem yang dibuat hanya bisa dilakukan dalam sistem penerjemah bahasa Russia ke struktur *data query*.

Penelitian yang dilakukan oleh Mohammad Kohar dapat melakukan perintah runtunan dan juga aritmatik dengan akurasi keberhasilan 98% [4]. Penelitian serupa juga dilakukan oleh Adi Yusuf yang membuat sistem yang penerjemah bahasa Indonesia ke *source code* dengan menggunakan Pascal. Pada penelitian tersebut, peneliti membuat sistem yang dapat menerima perintah runtunan, perintah perulangan, operasi aritmatika dasar dan operasi perbandingan dengan akurasi keberhasilan mencapai 95% [5]. Akan

tetapi, pada penelitian sebelumnya[4][5] tidak dapat menangani kesalahan pada masukan yang bersifat ambigu. Kesalahan yang ambigu dapat menyebabkan *error* sehingga pengguna tidak dapat mengetahui kesalahan pada masukan yang memicu *error* pada *Pascal*.

Berdasarkan uraian masalah yang telah disebut, penelitian ini akan mengembangkan sistem *Error Handling* pada sistem penerjemah dari bahasa alami dalam bahasa Indonesia ke source code kedalam bahasa *Pascal*. Pengembangan yang dimaksud adalah penambahan fitur *error handling* yang akan menampilkan reaksi dari teks inputan yang bersifat ambigu seperti kesalahan pada variabel dan kesalahan pada operator.

1.2 Rumusan Masalah

Berdasarkan latar belakang, rumusan masalah dalam penelitian ini yaitu bagaimana sistem penerjemah Natural Language (NL) berbahasa indonesia kedalam source code dapat mengatasi kesalahan pada teks masukan yang salah secara sintaks.

1.3 Maksud dan Tujuan

Berdasarkan masalah yang diteliti, maksud dari penelitian ini yaitu menambahkan sistem *Error Handling* pada sistem Penerjemah *Natural Language* (NL) Berbahasa Indonesia Kedalam *Source Code*.

Sedangkan tujuan dari penelitian ini yaitu agar dapat menangani kesalahan pada teks masukan yang salah secara sintaks pada sistem Penerjemah Natural Language (NL) Berbahasa Indonesia Kedalam Source Code.

1.4 Batasan Masalah

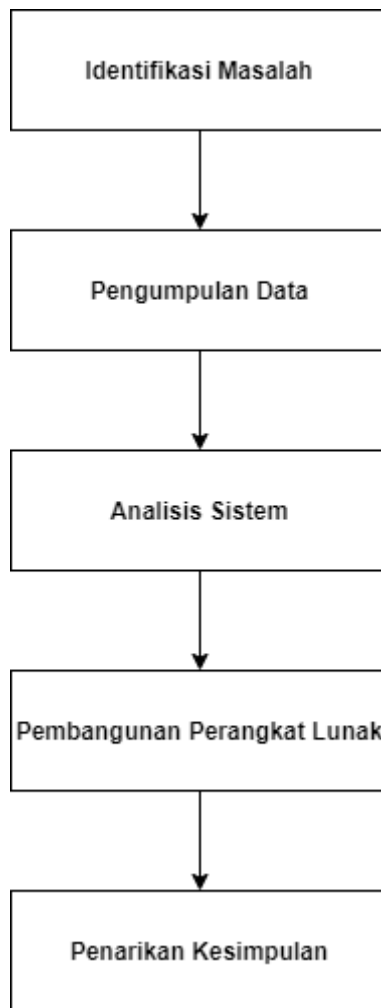
Menghindari penelitian yang meluas, maka penulis membatasi permasalahan sebagai berikut.

1. Data masukan berupa teks bahasa alami dalam bahasa Indonesia yang terstruktur dalam memecahkan masalah.
2. Bahasa alami yang dapat di translasi hanya perintah runtunan yang meliputi pembuatan variabel, pemanggilan fungsi *readln*, pemanggilan fungsi *writeln*, dan operasi matematik dasar (penjumlahan, pengurangan, perkalian, dan pembagian).
3. Proses
 1. Proses analisis yang akan ditambahkan adalah *Error Handling*.

4. Keluaran Data keluaran berupa *source code* runtunan dalam bahasa pemrograman Pascal dan nilai akurasi yang didapat dari proses penerjemahan.
5. *Error handling* yang akan diterapkan adalah penanganan kesalahan pada Semantik.
6. *Error handling* yang akan dibangun dibatasi hingga reaksi kesalahan.
7. *Error handling* tidak akan mengatasi kesalahan sintak lainnya seperti kesalahan sintak pada perulangan dan percabangan.

1.5 Metodologi Penelitian

Metode penelitian yang akan digunakan pada penelitian ini adalah metode penelitian kuantitatif. Metode penelitian kuantitatif adalah metode penelitian yang berguna untuk mendapatkan informasi berupa data yang dapat diukur. Metode penelitian meliputi metode pengumpulan data dan metode pembangunan perangkat lunak. Adapun alur penelitian yang terbagi menjadi 5 tahapan, yaitu identifikasi masalah, pengumpulan data, analisis sistem, pembangunan perangkat lunak, dan penarikan kesimpulan. Alur penelitian dapat dilihat pada Gambar 1.1.



Gambar 1.1 Alur Penelitian

1.5.1 Identifikasi Masalah

Tahap Identifikasi Masalah, merupakan tahap awal penelitian dimana pada proses ini dilakukan pengamatan terhadap penelitian sebelumnya supaya masalah dapat diukur (measurable).

1.5.2 Metode Pengumpulan Data

Untuk metode pengumpulan data yang dilakukan, antara lain:

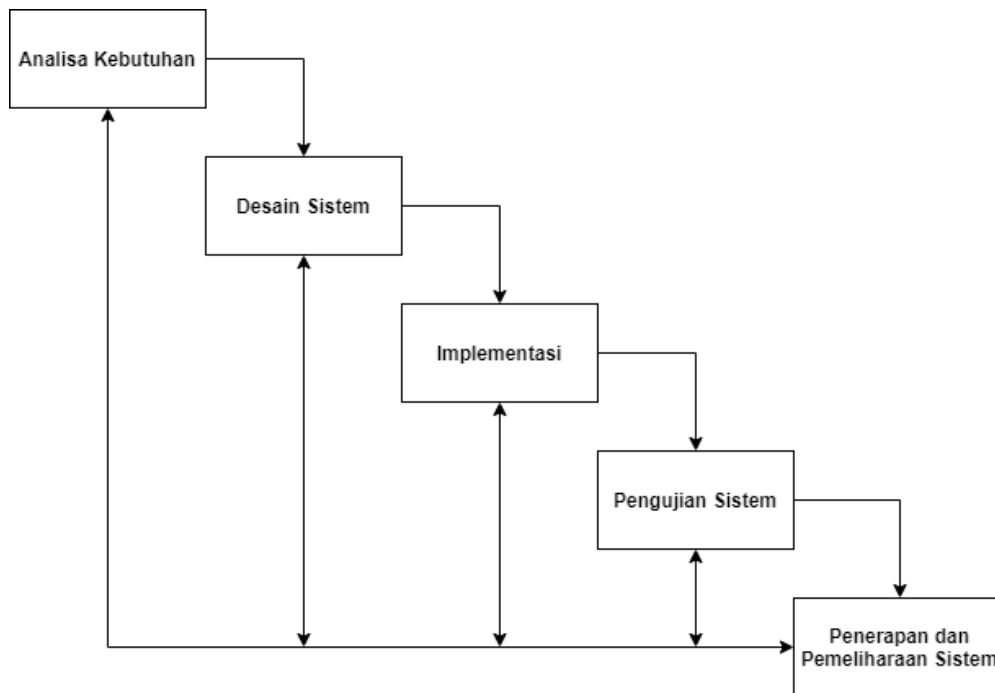
1. Pengamatan/Observasi
2. Studi Literatur.
3. Wawancara terstruktur.

1.5.3 Analisis Sistem

Pada tahap ini metode yang digunakan akan dianalisis. Metode yang digunakan dalam pembangunan sistem pada penelitian ini memiliki tiga tahapan yaitu *Preprocessing, Scanning, Parsing, Error handling*, dan proses translasi.

1.5.4 Pembangunan Perangkat Lunak

Metode pembangunan perangkat lunak yang digunakan pada penelitian ini menggunakan model *waterfall*.



Gambar 1.2 Metode Waterfall

1.5.4.1 Analisa Kebutuhan

Tahapan untuk menganalisis kebutuhan yang dibutuhkan oleh sistem dalam pembangunan perangkat lunak baik analisis kebutuhan fungsional maupun non fungsional.

1.5.4.2 Desain Sistem

Tahap perancangan antarmuka perangkat lunak.

1.5.4.3 Implementasi

Tahap mengimplementasikan desain kedalam bentuk program lengkap, dan uji coba untuk memastikan program sesuai dengan spesifikasi

1.5.4.4 Pengujian Sistem

Tahap unit program diuji coba sebagai sistem yang sudah lengkap.

1.5.4.5 Penerapan dan Pemeliharaan Sistem

Tahap pengurusan sistem setelah selesai dibuat dan ketika sedang dipakai untuk menjamin keberlangsungannya.

1.5.5 Penarikan Kesimpulan

Tahap Penarikan Kesimpulan, merupakan tahap dimana peneliti menarik kesimpulan berdasarkan dari hasil penelitian yang telah dilakukan.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian yang akan dibuat sebagai berikut

BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang, identifikasi masalah, maksud dan tujuan, batasan masalah, metode penelitian, serta sistematika penulisan dalam laporan penelitian ini.

BAB 2 LANDASAN TEORI

Bab ini membahas mengenai teori dasar yang digunakan untuk menganalisis masalah dan teori yang dipakai dalam penelitian, teori dasar yang dipakai dalam penelitian ini.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan secara teknis mengenai tahap-tahap yang dilakukan sistem *error handling* pada penerjemah bahasa alami dalam bahasa Indonesia ke source code dalam bahasa Pascal.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini menjelaskan mengenai implementasi dari proses analisis dan perancangan pada sistem *error handling* pada penerjemah bahasa alami dalam bahasa Indonesia ke source code dalam bahasa Pascal.

BAB 5 KESIMPULAN DAN SARAN

Bab ini menjelaskan hasil yang diperoleh dari tahap pengujian sistem *error handling* pada penerjemah bahasa alami dalam bahasa Indonesia ke source code dalam

bahasa Pascal. Hasil tersebut ditarik menjadi kesimpulan dan saran untuk penelitian selanjutnya.