

*'SOURCE-TO-SOURCE TRANSLATOR BAHASA  
PEMROGRAMAN C KE BAHASA PEMROGRAMAN  
JAVASCRIPT MENGGUNAKAN METODE RULE-BASED  
MACHINE TRANSLATION*

Diajukan Sebagai Syarat Untuk Menyelesaikan  
Pendidikan Program Strata-1 Pada  
Jurusan Teknik Informatika



Oleh:

M RYAN FADHOLI  
NIM: 09021281419050

**Jurusan Teknik Informatika  
FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA  
2018**

LEMBAR PENGESAHAN SKRIPSI

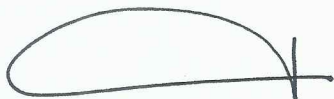
*SOURCE-TO-SOURCE TRANSLATOR BAHASA  
PEMROGRAMAN C KE BAHASA PEMROGRAMAN  
JAVASCRIPT MENGGUNAKAN METODE *RULE-BASED  
MACHINE TRANSLATION**

Oleh:

M RYAN FADHOLI  
NIM: 09021281419050

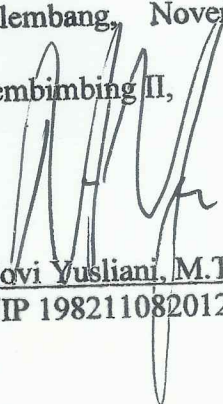
Palembang, November 2018

Pembimbing I



Drs. Megah Mulya, M.T  
NIP 196602202006041001

Pembimbing II,



Novi Yusliani, M.T  
NIP 198211082012122001

Mengetahui.  
Ketua Jurusan Teknik Informatika



Rifkie Primartha, M.T  
NIP 197706012009121004

## TANDA LULUS UJIAN SIDANG TUGAS AKHIR

Pada hari Jumat, 16 November 2018 telah dilaksanakan ujian sidang tugas akhir oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

Nama : M. Ryan Fadholi  
NIM : 09021281419050  
Judul : *Source-to-Source Translator Bahasa Pemrograman C ke Bahasa Pemrograman Javascript Menggunakan Metode Rule-Based Machine Translation*

### 1. Pembimbing I

Drs. Megah Mulya, M.T  
NIP. 197102041997021003

### 2. Pembimbing II

Novi Yusliani, M.T  
NIP. 198211082012122001

### 3. Penguji I

Yunita, M.Cs  
NIP. 198306062015042002

### 4. Penguji II

Kanda Januar Miraswan, M.T  
NIK. 1671080901900006



Mengetahui,  
Ketua Jurusan Teknik Informatika



Rifkie Primartha, M.T  
NIP. 197706012009121004

## HALAMAN PERNYATAAN BEBAS PLAGIAT

Yang bertanda tangan di bawah ini :

Nama : M. Ryan Fadholi  
NIM : 09021281419050  
Program Studi : Teknik Informatika  
Judul Skripsi : *Source-to-source Translator* Bahasa Pemrograman C ke  
Bahasa Pemrograman Javascript Menggunakan Metode  
*Rule-Based Machine Translation*  
Hasil Pengecekan Software *iThenticate/Turnitin* : 9 %

Menyatakan bahwa Laporan Proyek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan proyek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian, pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapapun.

Palembang, 30 November 2018



M. Ryan Fadholi  
NIM. 09021281419050

"Jika Kamu tidak dapat menahan lelahnya belajar, maka kamu harus sanggup menahan perihnya Kebodohan."

- **Imam Syafi'i**

"We are what we repeatedly do. Excellence, then, is not an act, but a habit."

- **Will Durant**

"Remember to look up at the stars and not down at your feet. Try to make sense of what you see and wonder about what makes the universe exist. Be curious. And however difficult life may seem, there is always something you can do and succeed at. It matters that you don't just give up."

- **Stephen Hawking**

Kupersembahkan Skripsi ini kepada:

1. Allah SWT
2. Kedua Orang Tuaku
3. Adikku
4. Keluarga Besar Teknik Informatika Unsri
5. Almamaterku

# A C PROGRAMMING LANGUAGE TO JAVASCRIPT PROGRAMMING LANGUAGE SOURCE-TO-SOURCE TRANSLATOR BASED ON RULE-BASED MACHINE TRANSLATION METHOD

By:  
**M. Ryan Fadholi**  
09021281419050

## ABSTRACT

As the field of computer science develops, existing programming languages may see a decrease in popularity and usage. When a programming language becomes obsolete it is necessary to do a code migration, an act of translating source code from one programming language to another. However, the code migration process tends to be very taxing and error-prone. Source-to-source translation is a possible solution to this problem, as it could translate source code automatically from one language to another. This research implements the rule-based machine translation method to build a source-to-source translation that accepts code written in C and translates it to Javascript. After the software is successfully built, it is then tested using 50 source code files written in C. The result shows that the software is able to translate the source codes with 100% accuracy. When the translation results was run, 34 or 68% of the translation result produces a similar output with the input source code. Analysis of translation results with differing output reveals common causes of these differences, namely the use of address-of (&) operators, the use of the rand() function, difference in arithmetic conversion rules, difference in non-initialized variables, and difference in the precision of format specifiers.

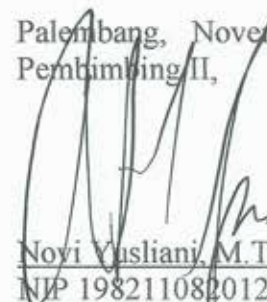
**Keywords:** Code Migration, C Programming Language, Javascript Programming Language, *Source-to-Source Translation, Rule-Based Machine Translation*

Pembimbing I



Drs. Megah Mulya, M.T  
NIP 196602202006041001

Palembang, November 2018  
Pembimbing II,



Novi Yusliani, M.T  
NIP 198211082012122001

Mengetahui.  
Ketua Jurusan Teknik Informatika



Rifkie Primartha, M.T  
NIP 197706012009121004

**SOURCE-TO-SOURCE TRANSLATOR BAHASA  
PEMROGRAMAN C KE BAHASA PEMROGRAMAN  
JAVASCRIPT MENGGUNAKAN METODE *RULE-BASED  
MACHINE TRANSLATION***

Oleh:  
**M. Ryan Fadholi**  
**09021281419050**

**ABSTRAK**

Seiring perkembangan ilmu komputer, bahasa-bahasa pemrograman yang ada dapat menurun popularitas dan penggunaannya. Saat sebuah bahasa pemrograman menjadi *obsolete*, terkadang diperlukan migrasi atau pemindahan *source code* dari satu bahasa pemrograman ke bahasa lain. Namun, proses migrasi kode secara manual cenderung rawan kesalahan serta menghabiskan waktu dan sumber daya. Salah satu solusi masalah tersebut adalah perangkat lunak *source-to-source translator* yang dapat menerjemahkan *source code* secara otomatis. Penelitian ini mengimplementasikan metode *rule-based machine translation* untuk membangun sebuah *source-to-source translator* bahasa pemrograman C ke bahasa pemrograman Javascript. Perangkat lunak *source-to-source translator* tersebut lalu diuji terhadap 50 *file source code* berbahasa C. Hasilnya perangkat lunak yang dibangun dapat melakukan translasi *source code* dengan akurasi 100%. Saat dijalankan, 34 atau 68% dari kode hasil translasi menghasilkan keluaran yang sama dengan kode masukannya. Hasil analisis dari kode dengan keluaran berbeda menunjukkan penyebab umum perbedaan tersebut, yaitu penggunaan *address-of (&) operator*, penggunaan fungsi *rand()*, perbedaan aturan *arithmetic conversion*, perbedaan pada variabel yang tidak diinisialisasi, dan perbedaan *precision* pada *format specifier*.

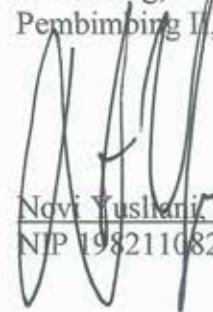
**Kata Kunci:** Migrasi Kode, Bahasa Pemrograman C, Bahasa Pemrograman Javascript, *Source-to-Source Translation*, *Rule-Based Machine Translation*

Pembimbing I



Drs. Megah Mulya, M.T  
NIP 196602202006041001

Palembang, November 2018  
Pembimbing II,



Novi Yushanji, M.T  
NIP 198211082012122001

Mengetahui.  
Ketua Jurusan Teknik Informatika



Rifkie Primartha, M.T  
NIP 197706012009121004.

# **A C PROGRAMMING LANGUAGE TO JAVASCRIPT PROGRAMMING LANGUAGE SOURCE-TO-SOURCE TRANSLATOR BASED ON RULE-BASED MACHINE TRANSLATION METHOD**

**By:**  
**M. Ryan Fadholi**  
**09021281419050**

## **ABSTRACT**

As the field of computer science develops, existing programming languages may see a decrease in popularity and usage. When a programming language becomes obsolete it is necessary to do a code migration, an act of translating source code from one programming language to another. However, the code migration process tends to be very taxing and error-prone. Source-to-source translation is a possible solution to this problem, as it could translate source code automatically from one language to another. This research implements the rule-based machine translation method to build a source-to-source translation that accepts code written in C and translates it to Javascript. After the software is successfully built, it is then tested using 50 source code files written in C. The result shows that the software is able to translate the source codes with 100% accuracy. When the translation results was run, 34 or 68% of the translation result produces a similar output with the input source code. Analysis of translation results with differing output reveals common causes of these differences, namely the use of address-of (&) operators, the use of the rand() function, difference in arithmetic conversion rules, difference in non-initialized variables, and difference in the precision of format specifiers.

**Keywords:** Code Migration, C Programming Language, Javascript Programming Language, *Source-to-Source Translation*, *Rule-Based Machine Translation*



***SOURCE-TO-SOURCE TRANSLATOR BAHASA  
PEMROGRAMAN C KE BAHASA PEMROGRAMAN  
JAVASCRIPT MENGGUNAKAN METODE *RULE-BASED  
MACHINE TRANSLATION****

**Oleh:  
M. Ryan Fadholi  
09021281419050**

**ABSTRAK**

Seiring perkembangan ilmu komputer, bahasa-bahasa pemrograman yang ada dapat menurun popularitas dan penggunaannya. Saat sebuah bahasa pemrograman menjadi *obsolete*, terkadang diperlukan migrasi atau pemindahan *source code* dari satu bahasa pemrograman ke bahasa lain. Namun, proses migrasi kode secara manual cenderung rawan kesalahan serta menghabiskan waktu dan sumber daya. Salah satu solusi masalah tersebut adalah perangkat lunak *source-to-source translator* yang dapat menerjemahkan *source code* secara otomatis. Penelitian ini mengimplementasikan metode *rule-based machine translation* untuk membangun sebuah *source-to-source translator* bahasa pemrograman C ke bahasa pemrograman Javascript. Perangkat lunak *source-to-source translator* tersebut lalu diuji terhadap 50 *file source code* berbahasa C. Hasilnya perangkat lunak yang dibangun dapat melakukan translasi *source code* dengan akurasi 100%. Saat dijalankan, 34 atau 68% dari kode hasil translasi menghasilkan keluaran yang sama dengan kode masukannya. Hasil analisis dari kode dengan keluaran berbeda menunjukkan penyebab umum perbedaan tersebut, yaitu penggunaan *address-of (&) operator*, penggunaan fungsi *rand()*, perbedaan aturan *arithmetic conversion*, perbedaan pada variabel yang tidak diinisialisasi, dan perbedaan *precision* pada *format specifier*.

**Kata Kunci:** Migrasi Kode, Bahasa Pemrograman C, Bahasa Pemrograman Javascript, *Source-to-Source Translation*, *Rule-Based Machine Translation*

## KATA PENGANTAR



Segala puji syukur kepada Allah SWT atas segala karunia, rahmat, dan kasih sayang-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “ABCD” dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan Pendidikan program Strata-1 pada Fakultas Ilmu Komputer Program Studi Teknik Informatika di Universitas Sriwijaya.

Perjalanan panjang telah penulis lalui dalam rangka perampungan penulisan skripsi ini. Banyak hambatan yang dihadapi dalam penyusunannya, namun berkat kehendak-Nyalah sehingga penulis berhasil menyelesaikan penyusunan skripsi ini. Oleh karena itu, dengan penuh kerendahan hati, pada kesempatan ini patutlah kiranya penulis mengucapkan terima kasih kepada:

1. Orang tuaku tercinta Bapak M. Alfian dan Ibu Erry Koriyanti, adikku Salwa Amalia, serta seluruh keluarga besarku yang selalu memberikan dukungan berharga berupa kasih sayang, didikan, nasihat, motivasi, dan doa.
2. Bapak Jaidan Jauhari, S.Pd., M.T. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
3. Bapak Rifkie Primartha, S.T., M.T. selaku Ketua Jurusan Teknik Informatika yang telah membimbing dan memberi motivasi kepada penulis dalam menyelesaikan skripsi ini.
4. Bapak Drs. Megah Mulya, M.T selaku Dosen Pembimbing I dan Ibu Novi Yusliani, M.T selaku Dosen Pembimbing II yang telah bersedia meluangkan

waktu dan tenaga serta memberi nasihat, saran, motivasi dan koreksi yang sangat berarti dalam membimbing penulis menyelesaikan skripsi ini.

5. Ibu Yunita, M.Cs. dan Bapak Kanda Januar Miraswan, M.T. selaku Dosen Penguji I dan II yang telah memberikan tanggapan dan saran yang bermanfaat dalam pengerjaan skripsi ini.
6. Bapak Osvari Arsalan, S.Kom., M.T. selaku pembimbing akademik yang telah memberikan arahan, masukan, dan nasihat serta memudahkan seluruh hal yang berkaitan dengan perkuliahan dan pengajuan judul tugas akhir ini.
7. Seluruh dosen Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya yang telah memberikan ilmunya kepada penulis selama proses perkuliahan.
8. Seluruh staf administrasi dan pegawai Fakultas Ilmu Komputer Universitas Sriwijaya yang telah blablabla
9. Seluruh teman-teman anggota IF Reguler 2014, terutama anggota kelas B dan perkumpulan Gua Griya yang telah menghabiskan waktu, menghibur, memotivasi, dan berjuang bersama Penulis semasa kuliah.
10. Kakak-kakak tingkat IF angkatan 2012 dan 2013, serta adik-adik tingkat angkatan 2015 dan 2016 yang telah banyak sekali membantu Penulis.
11. Teman-teman anggota BPH HMIF 2015 & 2016, iDev, serta organisasi-organisasi lain yang telah memberikan ruang bagi penulis untuk belajar dan berkarya.

12. Teman-teman dari masa SMA: M. Rio Zulmansyah, Andi Kurniawan, Rahmat Hidayat, Hersa Rizki Rakhmatullah, Vinny Ananda, Rien Larasati Arini untuk waktu, semangat dan bantuan yang diberikan kepada penulis.
13. Semua pihak yang telah membantu dalam penyelesaian tugas akhir ini dan tidak dapat disebutkan satu persatu. Semoga segala kebaikan yang diberikan mendapat balasan dari Allah SWT.

Penulis menyadari bahwa dalam penyusunan Skripsi ini masih terdapat banyak keukrangan disebabkan keterbatasan pengetahuan dan pengalaman, oleh karena itu kritik dan saran yang membangun sangat diharapkan untuk kemajuan penelitian selanjutnya.

Akhir kata, dengan segala kerendahan hati, semoga Skripsi ini dapat berguna dan bermanfaat bagi kita semua.

Indralaya, November 2018

Penulis

## DAFTAR ISI

Halaman

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN .....	ii
HALAMAN PERSETUJUAN KOMISI PENGUJI .....	iii
HALAMAN MOTTO DAN PERSEMBAHAN .....	iv
ABSTRACT .....	v
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	x
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR LAMPIRAN .....	xv

### BAB I PENDAHULUAN

1.1 Pendahuluan .....	1
1.2 Latar Belakang Masalah .....	1
1.3 Rumusan Masalah .....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	5
1.6 Batasan Masalah .....	5
1.7 Sistematika Penulisan .....	7
1.8 Kesimpulan .....	8

### BAB II KAJIAN LITERATUR

2.1 Pendahuluan .....	1
2.2 Landasan Teori .....	1
2.2.1 <i>Source-to-Source Translator</i> .....	1
2.2.2 <i>Rule-Based Machine Translation</i> .....	2
2.2.3 Bahasa Pemrograman C .....	4
2.2.4 Bahasa Pemrograman Javascript .....	6
2.3 Penelitian Lain yang Relevan .....	7
2.4 Kesimpulan .....	10

### BAB III METODOLOGI PENELITIAN

3.1 Pendahuluan .....	1
3.2 Metode Pengumpulan Data .....	1

3.3 Tahapan Penelitian .....	1
3.4 Metode Pengembangan Perangkat Lunak.....	5
3.4.1 Fase Insepsi.....	5
3.4.2 Fase Elaborasi.....	6
3.4.3 Fase Konstruksi.....	6
3.4.4 Fase Transisi .....	7
3.5 Manajemen Proyek Penelitian.....	7
3.6 Kesimpulan .....	15

#### BAB IV PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan .....	1
4.2 Fase Insepsi .....	1
4.2.1 Pemodelan Bisnis .....	1
4.2.2 Kebutuhan Sistem .....	2
4.2.3 Analisis dan Desain.....	4
4.3 Fase Elaborasi .....	15
4.3.1 Pemodelan Bisnis .....	15
4.3.2 Kebutuhan Sistem .....	17
4.3.3 Diagram <i>Sequence</i> .....	18
4.4 Fase Kontruksi.....	24
4.4.1 Kebutuhan Sistem .....	24
4.4.2 Diagram Kelas .....	25
4.4.3 Implementasi.....	27
4.5 Fase Transisi.....	30
4.5.1 Pemodelan Bisnis .....	30
4.5.2 Kebutuhan Sistem .....	31
4.5.3 Rencana Pengujian .....	31
4.5.4 Implementasi.....	32
4.6 Kesimpulan .....	35

#### BAB V HASIL DAN ANALISIS PENELITIAN

5.1 Pendahuluan .....	1
5.2 Data Hasil Percobaan/Penelitian .....	1
5.2.1 Konfigurasi Percobaan .....	1
5.2.2 Data Hasil Percobaan .....	4
5.3 Analisis Hasil Penelitian .....	6
5.3.1 Analisis Akurasi Hasil Translasi.....	6
5.3.2 Analisis Kesesuaian Keluaran Program Hasil Translasi ..	9
5.4 Kesimpulan .....	20

#### BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan .....	1
----------------------	---

6.2 Saran .....	1
DAFTAR PUSTAKA .....	xvi

## DAFTAR TABEL

Halaman

III-1. Tabel Hasil Translasi .....	4
III-2. Tabel Identifikasi Kesalahan pada Keluaran Program.....	5
III-3. Tabel Penjadwalan Penelitian dalam Bentuk <i>Work Breakdown Structure</i> (WBS) .....	8
IV-1. Tabel Kebutuhan Fungsional .....	4
IV-2. Tabel Kebutuhan Non Fungsional .....	4
IV-3. Contoh Hasil Proses Modul <i>Deformatter</i> .....	7
IV-4. Contoh Hasil Proses Modul <i>Part-of-Speech Tagger</i> .....	8
IV-5. Contoh Hasil Proses Modul <i>Structural &amp; Lexical Transfer</i> .....	9
IV-6. Definisi Aktor <i>Use Case</i> .....	11
IV-7. Deskripsi <i>Use Case</i> .....	11
IV-8. Skenario <i>Use Case</i> Menerjemahkan <i>Source Code</i> C ke Javascript.....	13
IV-9. Skenario <i>Use Case</i> Menyimpan <i>Source Code</i> Javascript Hasil Terjemahan.....	14
IV-10. Tabel Implementasi Kelas.....	27
IV-11. Rencana Pengujian <i>Use Case</i> Menerjemahkan <i>Source Code</i> Bahasa C ke Bahasa Javascript .....	32
IV-12. Pengujian <i>Use Case</i> Menerjemahkan <i>Source Code</i> Bahasa C ke Bahasa Javascript.....	33
V-1. Tabel Hasil Translasi .....	4
V-2. Tabel Rincian Jenis <i>Statement</i> yang Ditranslasi.....	7
V-3. Tabel Identifikasi Kesalahan pada Keluaran Program.....	9
V-4. Tabel Hasil Analisis Ketidaksesuaian Keluaran pada Program Hasil Translasi .....	10
V-5. Perbandingan Potongan Kode dan Keluaran dari Program Masukan dan Program Hasil Translasi (Data Uji <i>pointer_arg.c</i> ) .....	11
V-6. Perbandingan Potongan Kode dan Keluaran dari Program Masukan dan Program Hasil Translasi (Data Uji <i>bubble_sort.c</i> ) .....	12
V-7. Perbandingan Potongan Kode dan Keluaran dari Program Masukan dan Program Hasil Translasi (Data Uji <i>day_of_week.c</i> ) .....	14
V-8. Perbandingan Hasil Operasi Aritmatika pada Bahasa C dan Bahasa Javascript.....	16
V-9. Perbandingan Potongan Kode dan Keluaran dari Program Masukan dan Program Hasil Translasi (Data Uji <i>babbage_problem.c</i> ).....	17
V-10. Perbandingan Potongan Kode dan Keluaran dari Program Masukan dan Program Hasil Translasi (Data Uji <i>feigenbaum.c</i> ).....	18



## DAFTAR GAMBAR

Halaman

II-1. Arsitektur <i>Rule-Based Machine Translation</i> (Corbí-Bellot et al., 2005).....	3
II-2. Contoh Kode Sederhana dalam Bahasa C .....	5
II-3. Contoh Kode Sederhana dalam Bahasa Javascript .....	6
III-1. Diagram Tahap Pengolahan Data Masukan .....	2
III-2. Penjadwalan untuk Tahap Menentukan Ruang Lingkup dan Sumber Data Penelitian dan Tahap Menentukan Dasar Teori yang Berkaitan dengan Penelitian.....	12
III-3. Penjadwalan untuk Tahap Menentukan Tahapan Penelitian dan Tahap Melakukan Pengembangan Perangkat Lunak pada fase <i>Inception</i> dan <i>Elaboration</i> .....	13
III-4. Penjadwalan untuk Tahap Melakukan Pengembangan Perangkat Lunak pada fase <i>Construction</i> dan <i>Transition</i> , Tahap Melakukan Penelitian, dan Tahap Melakukan Analisa Hasil Pengujian dan Membuat Kesimpulan ....	14
IV-1. Alur <i>Source Translation</i> pada Perangkat Lunak .....	6
IV-2. Contoh Masukan Perangkat Lunak.....	6
IV-3. Contoh Hasil Proses Modul <i>Post-Generator</i> .....	10
IV-4. Diagram <i>Use Case</i> .....	11
IV-5. Diagram Aktivitas Menerjemahkan <i>Source Code</i> C ke Javascript .....	14
IV-6. Diagram Aktivitas Menyimpan <i>Source Code</i> Javascript Hasil Terjemahan.....	15
IV-7. Rancangan Antar Muka Perangkat Lunak .....	16
IV-8. Diagram <i>Sequence</i> Menyimpan <i>Source Code</i> Javascript Hasil Terjemahan.....	18
IV-9. Diagram <i>Sequence</i> Menerjemahkan <i>Source Code</i> C ke Javascript .....	19
IV-10. Diagram <i>Sequence</i> Menentukan <i>Statement</i> Berikutnya .....	20
IV-11. Diagram <i>Sequence</i> Menentukan Jenis <i>Statement</i> .....	21
IV-12. Diagram <i>Sequence</i> Menentukan Akhir <i>Statement</i> .....	22
IV-13. Diagram <i>Sequence</i> Menentukan Akhir <i>Set</i> Simbol Kurung.....	22
IV-14. Diagram <i>Sequence</i> Mencari <i>Substring</i> dalam Teks .....	23
IV-15. Diagram Kelas .....	26
IV-16. Antarmuka Utama Perangkat Lunak .....	28

## DAFTAR LAMPIRAN

1. TATA BAHASA PEMROGRAMAN C
2. TATA BAHASA PEMROGRAMAN JAVASCRIPT
3. DAFTAR TOKEN BAHASA PEMROGRAMAN C
4. DAFTAR TOKEN BAHASA PEMROGRAMAN JAVASCRIPT
5. ATURAN TRANSLASI PERANGKAT LUNAK
6. DOKUMENTASI *SOURCE CODE*
7. DOKUMENTASI *TEST CASE*

# **BAB I**

## **PENDAHULUAN**

### **1.1 Pendahuluan**

Bab ini membahas latar belakang masalah, rumusan masalah, tujuan dan manfaat penelitian serta batasan masalah. Bab ini juga akan memberikan penjelasan umum mengenai keseluruhan penelitian.

Pendahuluan dimulai dengan penjelasan singkat tentang bahasa pemrograman. Selanjutnya bab ini akan membahas kebutuhan dan tantangan melakukan migrasi kode antar bahasa pemrograman yang menjadi latar belakang masalah penelitian ini, serta translasi otomatis sebagai solusinya.

### **1.2 Latar Belakang Masalah**

Bahasa pemrograman adalah bahasa tingkat tinggi yang mudah dimengerti manusia, dan digunakan sebagai pengganti bahasa mesin bagi manusia untuk memberikan perintah untuk komputer. Agar dapat dijalankan oleh komputer, perintah yang telah ditulis dalam bahasa pemrograman lalu diproses oleh *compiler* menjadi bahasa mesin.

Seiring dengan perkembangan ilmu komputer, banyak bermunculan bahasa-bahasa pemrograman dengan fitur-fitur dan keuntungan yang beragam. Saat ini ada ratusan bahasa pemrograman dengan fungsi, kelebihan, dan kekurangan masing-masing.

Salah satu bahasa pemrograman yang saat ini paling banyak digunakan dan diajarkan baik di lingkungan industri dan lingkungan akademis ilmu komputer di Indonesia adalah bahasa pemrograman C. Alasan penggunaan bahasa C antara lain adalah penggunaan RAM yang efisien, ukuran program *native* hasil kompilasi yang kecil, dan waktu eksekusi yang cepat (Nanz & Furia, 2015).

Namun secara umum tren penggunaan bahasa C semakin lama semakin menurun<sup>1)</sup>. Hal ini disebabkan oleh munculnya bahasa-bahasa baru dengan fitur yang lebih modern, bertambahnya *library*, bertambahnya referensi untuk bahasa-bahasa baru tersebut, dll. Selain itu mahasiswa merasa sulit mengerti konsep-konsep yang berhubungan dengan *memory* (Milne & Rowe, 2002). Bahasa C justru banyak menggunakan konsep yang berhubungan dengan *memory* seperti *pointer*.

Jika popularitas sebuah bahasa terus menurun hingga bahkan dianggap *obsolete*, maka terkadang perlu dilakukan migrasi kode program dalam bahasa tersebut ke bahasa lain yang lebih aktif digunakan dan dikembangkan. Hal ini dapat disebabkan oleh berbagai hal, antara lain perubahan dalam proses bisnis, integrasi, peraturan perusahaan, adaptasi teknologi baru, perubahan hardware, dan lain sebagainya.

Saat ini salah satu bahasa pemrograman dengan pertumbuhan penggunaan yang paling stabil adalah bahasa Javascript. Pada tahun 2016 pertumbuhan jumlah penggunaan bahasa Javascript pada proyek-proyek *open source* yang dipublikasikan di GitHub meningkat sebesar 97%<sup>2)</sup>. Saat ini bahasa Javascript umumnya digunakan

---

<sup>1)</sup> TIOBE software BV. "TIOBE Index." TIOBE – The Software Quality Company. <http://www.tiobe.com/tiobe-index/> (diakses 7 Maret 2017)

<sup>2)</sup> GitHub Inc. "GitHub Octoverse 2016." Github. <https://octoverse.github.com/> (diakses 7 Maret 2017)

pada program-program berbasis web seperti *website*, *webapp*, dan *webserver*. Selain itu, bahasa Javascript banyak digunakan sebagai bahasa perkenalan untuk pengajaran pemrograman (Ward & Smith, 1998; Mahmoud *et al.*, 2004).

Namun migrasi dari satu bahasa pemrograman ke bahasa yang lain tidaklah mudah. Untuk melakukan migrasi secara manual dibutuhkan penguasaan bahasa sumber dan tujuan (Qiu, 1999). Jika kode sumber aplikasi yang akan diterjemahkan kompleks dan memiliki ukuran yang besar, proses migrasi akan menghabiskan banyak waktu dan sumber daya, serta rawan kesalahan.

Salah satu solusi untuk melakukan migrasi antar bahasa pemrograman dengan lebih efisien adalah dengan menggunakan *source-to-source translator*, yang dapat digunakan untuk membantu migrasi dengan cara menerjemahkan kode dari satu bahasa ke bahasa lain secara otomatis (Plaisted, 2013).

Salah satu metode yang dapat digunakan untuk melakukan translasi otomatis antar bahasa pemrograman adalah *rule-based machine translation*. Menurut Forcada *et al.* (2011) *rule-based machine translation* memiliki karakteristik hasil translasi yang repetitif dan konsisten, sehingga cocok untuk digunakan untuk translasi bahasa pemrograman yang memiliki grammar jelas dan konsisten.

Berdasarkan uraian tersebut, penelitian ini akan mengimplementasikan metode *rule-based machine translation* untuk membangun sebuah *source-to-source translator* bahasa pemrograman C ke bahasa pemrograman Javascript sehingga migrasi kode dapat dilakukan secara otomatis.

### 1.3 Rumusan Masalah

Masalah yang diangkat pada penelitian ini adalah apakah metode *rule-based machine translation* dapat diimplementasikan untuk *source-to-source translator* bahasa pemrograman C ke bahasa pemrograman Javascript?

Dari masalah utama tersebut lalu diuraikan pertanyaan penelitian berikut:

1. Bagaimana cara mengimplementasikan metode *rule-based machine translation* yang telah ditentukan menjadi sebuah *source-to-source translator* bahasa pemrograman C ke bahasa pemrograman Javascript?
2. Bagaimana akurasi kode hasil terjemahan *source-to-source translator* yang menggunakan metode *rule-based machine translation* dalam menerjemahkan kode dalam bahasa pemrograman C ke bahasa pemrograman Javascript?

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menghasilkan sebuah program *source-to-source translator* bahasa pemrograman C ke bahasa pemrograman Javascript menggunakan metode *rule-based machine translation*.
2. Mengetahui akurasi hasil terjemahan *source-to-source translator* menggunakan metode *rule-based machine translation* dalam menerjemahkan kode bahasa pemrograman C ke bahasa pemrograman Javascript.

## 1.5 Manfaat Penelitian

Manfaat penelitian ini adalah:

1. Hasil penelitian dapat digunakan untuk membantu mahasiswa/pelajar yang telah menguasai bahasa pemrograman C dalam mempelajari bahasa pemrograman Javascript.
2. Hasil penelitian dapat digunakan sebagai rujukan untuk penelitian-penelitian *source-to-source translation* selanjutnya.

## 1.6 Batasan Masalah

Adapun batasan masalah penelitian ini adalah sebagai berikut:

1. Kode dalam bahasa C yang menjadi masukan adalah kode yang dapat di-*compile* oleh *compiler gcc* tanpa *compilation error*.
2. Masukan yang akan diterjemahkan adalah kode dari bahasa pemrograman C berdasarkan standar ANSI C.
3. Keluaran dari program ini adalah kode bahasa pemrograman Javascript berdasarkan standar ES6 yang diimplementasikan pada perangkat lunak Node.js.
4. Kode masukan dan keluaran adalah kode dengan paradigma prosedural.
5. Pada bahasa masukan, program hanya mendukung variabel *signed* dengan tipe data sebagai berikut:
  - a. Tipe data dasar (*base data types*):
    - i. *Character* (char)
    - ii. *Double* (double)

- iii. *Floating-Point* (float)
- iv. *Integer* (int)
- b. Tipe data tambahan (*modifier data types*):
  - i. *Long* (long)
  - ii. *Short* (short)
- 6. Dalam bahasa masukan, program mendukung *format specifier* %c, %d, %f, %hi, %i, %li, %lli dalam *format string* pada fungsi *printf* dan *scanf*.
- 7. Tipe-tipe *statement* dan *expression* dari bahasa masukan yang dapat ditranslasi adalah:
  - a. Deklarasi, inialisasi, dan inisiasi variabel.
  - b. Deklarasi, inialisasi, dan inisiasi *array* satu dimensi.
  - c. Deklarasi dan pemanggilan fungsi dan prosedur.
  - d. Penggunaan *pointer* sebagai *parameter* pada deklarasi dan pemanggilan fungsi.
  - e. Ekspresi aritmatika menggunakan operator +, -, \*, dan /.
  - f. Ekspresi *equality* (kesamaan) dan *inequality* (ketidaksamaan) dimana nilai/variabel yang dibandingkan mempunyai tipe data yang sama.
  - g. Masukan untuk satu variabel tanpa *format* tambahan menggunakan *keyword scanf* melalui *stdin*.
  - h. Keluaran menggunakan *keyword printf* melalui *stdout*.
  - i. Pengkondisian menggunakan *keyword if, else if, dan else*.
  - j. Pengulangan menggunakan *keyword for, while, dan do-while*.
  - k. Komentar *single-line* dan *multi-line*.



8. Program tidak mendukung data masukan berupa tipe data/struktur data bentukan seperti *linked list*, pohon (*tree*), tumpukan (*stack*), dan antrian (*queue*).

## **1.7 Sistematika Penulisan**

Sistematika penulisan skripsi ini adalah sebagai berikut:

### **BAB I. PENDAHULUAN**

Bab I menguraikan latar belakang dan rumusan masalah, tujuan dan manfaat penelitian, batasan masalah, dan sistematika penulisan penelitian ini.

### **BAB II. KAJIAN LITERATUR**

Bab II berisi landasan teori yang digunakan pada penelitian ini seperti *source-to-source translator*, *rule-based machine translation*, bahasa pemrograman C, dan bahasa pemrograman Javascript. Selain itu bab II juga membahas penelitian-penelitian lain yang relevan dengan penelitian ini.

### **BAB III. METODOLOGI PENELITIAN**

Bab III berisi pembahasan mengenai tahapan yang akan dilaksanakan pada penelitian ini. Rencana tahapan penelitian akan dideskripsikan dengan rinci dengan mengacu pada suatu kerangka kerja. Di akhir bab III akan dijabarkan perancangan manajemen proyek untuk pelaksanaan penelitian ini.

## **BAB IV. PENGEMBANGAN PERANGKAT LUNAK**

Bab IV berisi pembahasan mengenai proses perancangan dan lingkungan implementasi perangkat lunak *source-to-source translator* menggunakan metode *rule-based machine translation* pada penelitian ini.

## **BAB V. HASIL DAN ANALISIS PENELITIAN**

Bab V berisi penyajian hasil pengujian penelitian berdasarkan langkah-langkah yang telah direncanakan. Selain itu, di dalam bab ini juga dijabarkan analisis terhadap hasil pengujian sebagai basis dari kesimpulan yang diambil dalam penelitian ini.

## **BAB VI. KESIMPULAN DAN SARAN**

Bab VI berisi kesimpulan dari semua uraian-uraian pada bab-bab sebelumnya dan juga berisi saran-saran yang diharapkan berguna dalam penerapan dan penelitian *source-to-source translator* menggunakan metode *rule-based machine translation* selanjutnya.

### **1.8 Kesimpulan**

Pada bab ini telah dibahas latar belakang masalah penelitian ini dalam melakukan migrasi kode dari bahasa pemrograman C ke bahasa pemrograman Javascript. Karena itu, penelitian ini akan mengimplementasikan metode *rule-based machine translation* untuk membangun sebuah *source-to-source translator* dari kode bahasa pemrograman C ke bahasa pemrograman Javascript.

## DAFTAR PUSTAKA

- Adam, I.F. 2008. Aplikasi Translator Kode Dari Bahasa C ke Pascal.
- Corbí-Bellot, A.M., Forcada, M.L., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Alegria, I., Mayor, A. & Sarasola, K. 2005. An open-source shallow-transfer machine translation engine for the romance languages of Spain. Proceedings of the European Association for Machine Translation, 10th Annual Conference, 79–86. (<http://transducens.dlsi.ua.es/repositori/transducens/pubs/165/corbi05.pdf>).
- European Computer Manufacturers Association 2016. ECMAScript® 2016 Language Specification. (<https://www.ecma-international.org/publications/standards/Ecma-262.htm>).
- Forcada, M.L., Ginestí-Rosell, M., Nordfalk, J., O’regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Sánchez-Martínez, F., Ramírez-Sánchez, G. & Tyers, F.M. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Mach Translat*, 25: 127–144.
- Hutchins, J. 1994. Machine translation: History and general principles. *The encyclopedia of languages and linguistics*, 5: 2322–2332.
- Ilyushin, E. & Namiot, D. 2016. On source-to-source compilers. *International Journal of Open Information Technologies*, 4(5): 48–51.
- Kernighan, B.W. & Ritchie, D.M. 2006. *The C programming language*. Upper Saddle River: Prentice-Hall, Inc.
- Khalafinejad, S. & Mirian-Hosseiniabadi, S.H. 2010. Rule-based translation of specifications to executable code. *ICIME 2010 - 2010 2nd IEEE International Conference on Information Management and Engineering*, 1: 1–4.
- Mahmoud, Q.H., Dobosiewicz, W. & Swayne, D. 2004. Redesigning introductory computer programming with HTML, JavaScript, and Java. Proceedings of the 35th SIGCSE technical symposium on Computer science education - SIGCSE ’04, 120. (<http://portal.acm.org/citation.cfm?doid=971300.971344>).
- Milne, I. & Rowe, G. 2002. Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, 7(1): 55–66.
- Nanz, S. & Furia, C.A. 2015. A comparative study of programming languages in rosetta code. *Proceedings - International Conference on Software Engineering*, 1(978): 778–788.
- Plaisted, D.A. 2013. Source-to-Source Translation and Software Engineering. *Journal of Software Engineering and Applications*, 2013(April): 30–40.
- Prabowo, R. 2016. Implementasi Metode Rule Based Machine Translation Untuk Menerjemahkan Bahasa Pemrograman C++ ke Bahasa Pemrograman Python. Skripsi Program Teknik Informatika FAKULTAS ILMU KOMPUTER Palembang (tidak dipublikasikan).
- Qiu, L. 1999. *Programming Language Translation*. Cornell University.
- Tripathi, S. & Sarkhel, J.K. 2010. Approaches to machine translation. *Annals of Library and Information Systems*, 57(December): 388–393.

- Trudel, M., Furia, C.A., Nordio, M., Meyer, B. & Oriol, M. 2012. C to O-O translation: Beyond the easy stuff. Proceedings - Working Conference on Reverse Engineering, WCRE, 19–28.
- Ward, R. & Smith, M. 1998. JavaScript as a first programming language for multimedia students. *ACM SIGCSE Bulletin*. hal.249–253.
- Zakai, A. 2011. Emscripten: an LLVM-to-JavaScript Compiler. ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion, 301–312. (<http://doi.acm.org/10.1145/2048147.2048224>).