

**IMPLEMENTASI MODEL *STATE MACHINE* DALAM  
PENILAIAN PENULISAN *SOURCE CODE* OTOMATIS  
PADA SISTEM *ONLINE JUDGE***

**Skripsi**

*Diajukan untuk memenuhi bagian dari syarat*

*memperoleh gelar Sarjana Komputer*

*Program Studi Ilmu Komputer*



Oleh :

**Albari Berki Pradhana**

**1806384**

**PROGRAM STUDI ILMU KOMPUTER  
DEPARTEMEN PENDIDIKAN ILMU KOMPUTER  
FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN  
ALAM  
UNIVERSITAS PENDIDIKAN INDONESIA  
2022**

**IMPLEMENTASI MODEL *STATE MACHINE* DALAM  
PENILAIAN PENULISAN *SOURCE CODE* OTOMATIS  
PADA SISTEM *ONLINE JUDGE***

Oleh

Albari Berki Pradhana

NIM 1806384

Sebuah skripsi yang diajukan untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana pada Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam

© Albari Berki Pradhana

Universitas Pendidikan Indonesia

Agustus 2022

Hak cipta dilindungi Undang-Undang

Skripsi ini tidak boleh diperbanyak seluruhnya atau sebagian,  
dengan dicetak ulang, difotokopi, atau cara lainnya tanpa izin dari penulis.

**LEMBAR PENGESAHAN**  
**IMPLEMENTASI MODEL *STATE MACHINE* DALAM**  
**PENILAIAN PENULISAN *SOURCE CODE* OTOMATIS**  
**PADA SISTEM *ONLINE JUDGE***

Oleh

Albari Berki Pradhana

1806384

Disetujui dan disahkan oleh:

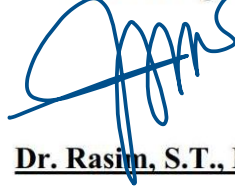
Pembimbing I,



**Rosa Ariani Sukanato, M. T.**

NIP. 198109182009122003

Pembimbing II,



**Dr. Rasim, S.T., M.T.**

NIP. 197407252006041002

Mengetahui,

Ketua Program Studi Ilmu Komputer



**Dr. Rani Megasari, M.T.**

NIP. 198705242014042002

**IMPLEMENTASI MODEL *STATE MACHINE* DALAM  
PENILAIAN PENULISAN *SOURCE CODE* OTOMATIS  
PADA SISTEM *ONLINE JUDGE***

Oleh

Albari Berki Pradhana - barbar18pradh@upi.edu

1806384

**ABSTRAK**

Proses penilaian *source code* dapat menjadi pekerjaan yang cukup berat. Salah satu penyebabnya adalah banyaknya jumlah *source code* yang perlu dinilai ditambah lagi dengan sedikitnya waktu yang tersedia untuk menyelesaikan penilaian seluruh *source code* tersebut. Sehingga kehadiran modul penilaian *source code* otomatis akan sangat membantu. Pada penelitian ini model *state machine* akan diimplementasikan kedalam modul penilaian otomatis. Dengan mengimplementasikan model *state machine* yang sesuai dengan kriteria penilaian, diharapkan modul penilaian otomatis yang dibangun akan melakukan penilaian selayaknya penilai. Hasil dari penelitian ini menunjukkan bahwa modul penilaian penulisan *source code* otomatis yang dibangun memiliki nilai koefisien korelasi yang cukup tinggi, yaitu 0,851 dengan skala 1. Namun, penelitian dan pengembangan lebih lanjut akan sangat diperlukan untuk membangun modul penilaian penulisan *source code* otomatis yang lebih baik.

Kata Kunci: *State Machine, CSPC, Source Code, Assessment, Online Judge, Correlation Coefficient*

***IMPLEMENTING STATE MACHINE MODEL IN  
AUTOMATIC SOURCE CODE WRITING  
ASSESSMENT ON ONLINE JUDGE***

*By*

Albari Berki Pradhana - barbar18pradh@upi.edu

1806384

**ABSTRACT**

*Source code assessment process can become a hard job. One of the reasons are the amount of source code they need to assessed are quiet a lot. More over, they won't have that much of a time to assessed that much of source code. That is why an existence of automatic source code writing assessment module will helped a lot as lecturer assistant or as a lecturer. In this research, state machine model will be implemented to the automatic assessment module. With implementing the correct state machine model that refers to the lecturer assistant's or lecturer's assessment criteria, the automatic assessment module will assessed all the source code as good as how the lecturer assistant or lecturer would assessed. The result of this research show that the automatic source code writing assessment module have a correlation coefficient value of 0,851 in a scale of 1, which is quiet high. But a future development and research will be needed to get a better automatic source code writing assessment module.*

Keywords: *State Machine, CSPC, Source Code, Assessment, Online Judge, Correlation Coefficient*

## DAFTAR ISI

KATA PENGANTAR.....	i
UCAPAN TERIMAKASIH.....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan.....	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika.....	5
BAB II.....	7
2.1 Source Code.....	8
2.2 Online Judge.....	17
2.3 Sistem Penilaian Otomatis.....	22
2.4 State Machine.....	23
2.5 Nilai Koefisien Korelasi.....	28
2.6 Penelitian terkait.....	28

BAB III.....	31
3.1  Desain Penelitian.....	31
3.1.1  Studi Literatur.....	32
3.1.2  Pengumpulan Data.....	32
3.1.3  Implementasi Metode.....	32
3.1.4  Analisis dan Evaluasi.....	33
3.1.5  Dokumentasi.....	34
3.2  Pengembangan Aplikasi.....	34
3.3  Alat dan Bahan penelitian.....	35
3.3.1  Alat Penelitian.....	35
3.3.2  Bahan Penelitian.....	36
3.3.3  Populasi dan Sampel Penelitian.....	36
3.4  Instrumen Penelitian.....	36
BAB IV.....	37
4.1  Kriteria Penilaian.....	37
4.2  Rancangan model state machine.....	39
4.3  Membangun Aplikasi.....	41
4.3.1  Desain Arsitektur.....	42
4.3.2  Membangun Database.....	42
4.3.3  Menerapkan State Machine.....	46
4.3.3.1  Penerapan Pada Indentasi.....	46
4.3.3.2  Penerapan Pada Semicolon.....	47
4.3.3.3  Penerapan Pada Parentheses.....	48

4.3.3.4 Penerapan Pada For.....	49
4.3.3.5 Penerapan Pada While.....	50
4.3.3.6 Penerapan Pada Do While.....	52
4.3.3.7 Penerapan Pada If Else.....	53
4.3.4 Menerapkan Sistem Bobot Penilaian.....	54
4.3.4.1 Penilaian Indentasi.....	54
4.3.4.2 Penilaian Semicolon.....	55
4.3.4.3 Penilaian Parentheses.....	56
4.3.4.4 Penilaian For.....	57
4.3.4.5 Penilaian While.....	58
4.3.4.6 Penilaian Do While.....	60
4.3.4.7 Penilaian If Else.....	61
4.3.4.8 Penilaian Total Nilai.....	62
4.3.5 Membangun Antarmuka.....	63
4.4 Validasi.....	68
4.5 Pengujian.....	75
4.6 Pembahasan.....	84
BAB V.....	87
5.1 Kesimpulan.....	87
5.2 Saran.....	87
DAFTAR PUSTAKA.....	89
LAMPIRAN.....	92



## DAFTAR PUSTAKA

- Wasik, S., Antczak, M., Badura, J., Laskowski, A., & Sternal, T. (2018). A Survey on Online Judge Systems and Their Applications. *ACM Computing Surveys*, 51(1), 1–34. <https://doi.org/10.1145/3143560>
- Kurnia, A., Lim, A., & Cheang, B. (2001). Online Judge. *Computers & Education*, 36(4), 299–315. [https://doi.org/10.1016/s0360-1315\(01\)00018-5](https://doi.org/10.1016/s0360-1315(01)00018-5)
- Mekterovic, I., Brkic, L., Milasinovic, B., & Baranovic, M. (2020). Building a Comprehensive Automated Programming Assessment System. *IEEE Access*, 8, 81154–81172. <https://doi.org/10.1109/access.2020.2990980>
- R A Sukamto et al 2017 *J. Phys.: Conf. Ser.* 812 012109
- Development for Learning Programming. *Journal of Physics: Conference Series*. 812. 012109. [10.1088/1742-6596/812/1/012109](https://doi.org/10.1088/1742-6596/812/1/012109).
- Rischa, F., Sukamto, R., & Piantari, E. (2021). Perhitungan Text Similarity Berbasis Word Embedding dengan Word Mover Distance untuk Penilaian Komentar Otomatis dalam Sistem Online Judge - UPI Repository. *Repository.Upi.Edu*. <http://repository.upi.edu/59325/>
- Sukamto, R. A. (2011). CSPC. <http://cspc.cs.upi.edu/>
- Buse, R. P. L., & Weimer, W. R. (2010). Learning a Metric for Code Readability. *IEEE Transactions on Software Engineering*, 36(4), 546–558. <https://doi.org/10.1109/ICSE-COMPANION.2009.5070980>
- Lientz, B. P. (1983). Issues in Software Maintenance. *ACM Computing Surveys*, 15(3), 271–278. <https://doi.org/10.1145/356914.356919>
- Olivya, Meylanie. (2015). SISTEM PENILAI SOURCE CODE OTOMATIS.
- K.Danutama and I. Liem, “Scalable Autograder and LMS Integration,” *Procedia Technol.*, vol. 11, pp. 388–395, 2013.
- Jan A. Bergstra; Alban Ponse (2002). Combining programs and state machines. , 51(2), 175–192. doi:10.1016/s1567-8326(02)00020-6

- Miftah F. R., Addy S., Septya M. (2016). PENERAPAN METODE FINITE STATE MACHINE PADA GAME “THE RELATIONSHIP”. Mulawarman University <http://e-journals.unmul.ac.id/index.php/JIM/index>
- Ruslan, Gunawan, Suhatati T. (2018). Sistem Penelitian Otomatis Jawaban Esai Menggunakan Metode GLSA. Seminar Nasional Aplikasi Teknologi Informasi (SNATi) 2018. ISSN: 1907 – 5022
- TIOBE the software quality company Programming Community index (2009). “TIOBE PROGRAMMING COMMUNITY INDEX”, <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- Dr. Suarga, 2012 “Algoritma dan Pemrograman”, Penerbit Andi, Yogyakarta, Kusumadewi, 2010, Artificial Intelligence (Teknik dan Aplikasinya), Yogyakarta: Graha Ilmu.
- Taufik Pratama M., (2014). EVOLUSI BAHASA PEMROGAMAN (*Evolution of Programming Language*). Jurnal Computech & Bisnis, Vol. 8, No1, 25042 ISSN : 2442-4943
- Bolton, David. (2021, February 16). Definition of Source Code. Retrieved from <https://www.thoughtco.com/source-code-definition-958200>
- Muhammad Wali, Lukman Ahmad., (2017). Perancangan Aplikasi *Source code library* Sebagai Solusi Pembelajaran Pengembangan perangkat lunak. Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi) 1 (1) : <http://journal.lembagakita.org/index.php/jtik>
- Kufner, Josef; Marik, Radek (2019). Restful State Machines and SQL Database. IEEE Access, 7(), 144603–144617. doi : 10.1109/access.2019.2944807
- Brian W. Kernighan and Dennis M. Ritchie. 1988. *The C Programming Language* (2nd. ed.). Prentice Hall Professional Technical Reference.
- LaMeres, B. J. (2019). Modeling Finite State Machines. Quick Start Guide to VHDL, 127–141. doi:10.1007/978-3-030-04516-6\_9

- Telussa, Ade M., Elvinus R. Persulesy, and Zeth A. Leleury. "Penerapan analisis korelasi parsial untuk menentukan hubungan pelaksanaan fungsi manajemen kepegawaian dengan efektivitas kerja pegawai." *BAREKENG: Jurnal Ilmu Matematika dan Terapan* 7.1 (2013):15-1
- Pressman, Roger S.(2001). *Software engineering: a practitioner's approach*—5th ed. New York: McGraw-Hill Publishing Company, Inc.
- Sukamto, R. A., Megasari, R., Piantari, E., & Rischa, M. N. F. (2020). Code Comment Assessment Development for Basic Programming Subject using Online Judge. In *Proceedings of the 7th Mathematics, Science, and Computer Science Education International Seminar, MSCEIS 2019*. European Alliance for Innovation. <https://doi.org/10.4108/eai.12-10-2019.2296547>
- Douce, C., Livingstone, D., & Orwell, J. (2005). Automatic test-based assessment of programming. *Journal on Educational Resources in Computing*, 5(3), 4–es. doi:10.1145/1163405.116340

