|  |  |
| --- | --- |
|  | Title is in english max 15 words left alignment bold arial 14 sentence case |

[](https://crossmark.crossref.org/dialog/?doi=10.22219/kinetik.v4i3.841&domain=pdf)

First Author\*1, Second Author2, Third Author3

1, First Afiliation, Country

2, Second Afiliation, Country

3, Third Afiliation, Country

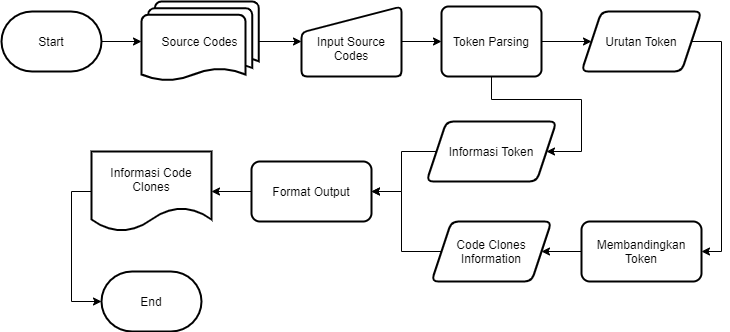
|  |  |
| --- | --- |
| Article Info | Abstract |
| Keywords:  Internet of Things Platform, Internet of Things, Message Queuing Telemetry Transport, MQTT Broker Server  Article history:  Received 17 August 2018  Revised 15 February 2019  Accepted 4 April 2019  Available online 4 April 2019  Cite:  Wardana, A., Rakhmatsyah, A., Minarno, A., & Anbiya, D. (2019). Internet of Things Platform for Manage Multiple Message Queuing Telemetry Transport Broker Server. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 4(3). doi:<http://dx.doi.org/10.22219/kinetik.v4i3.841>  \* Corresponding author.  Corresponding Author  E-mail address:  author@email.ac.id | *Code clones merupakan segmen kode di dalam suatu source files yang identik atau memiliki kemiripan antara satu sama lain. Semakin besar dan kompleks suatu sistem, semakin banyak kebutuhan sumber daya dan biaya yang diperlukan untuk mendeteksi code clones. Salah satu cara untuk meringankan sumber daya pada proses mendeteksi code clones adalah dengan otomasi proses mendeteksi code clones. Pada penelitian ini, proses mendeteksi code clones secara otomatis yang akan dirancang menggunakan pendekatan token. Penelitian ini bertujuan untuk mengevaluasi sebuah sistem pendeteksi code clones otomatis menggunakan pendekatan token untuk mengobservasi keakuratan hasil dan mengamati efisiensi waktu sistem tersebut.* |
|  |  |

1. Introduction
2. Research Method
   1. Datasets

Penelitian ini menggunakan hasil dari penelitian A Code Clone Oracle yang diterbitkan pada tahun 2014 oleh MSR 2014.

* 1. Perancangan Sistem

Sistem yang akan dirancang menerima inputan berupa source code dari satu atau lebih program. Sistem yang akan dirancang memiliki lima tahap utama yaitu input source codes, token parsing, string matching, format output dan evaluate result. merupakan alur dari sistem pendeteksi code clones.



* + 1. Input Source Code

Pada tahap ini dilakukan pembacaan file *source code* yang menjadi input pada sistem ini. Input dapat berupa aplikasi dari *source code Python*. Sistem dapat menerima input dari beberapa *source codes* maupun satu *source code*. Jika sistem menerima banyak inputan *source codes* maka sistem akan menyisipkan ke satu buah file sehingga terbaca seperti satu buah *source code*.



* + 1. Token Parsing

Figure 3‑4 merupakan tiga tahap utama dalam mengubah inputan *source codes* menjadi sebuah *token sequence* dan informasi baris fungsi/prosedur dari token tersebut. Tahapan ini dibagai menjadi tiga bagian, yaitu *remove white space*, *shorten names*, dan *tokenize*. Hasil proses ini adalah hasil transformasi *token sequence* informasi seluruh fungsi dari *token sequence*.



* + 1. Remove White Space And Comment

Tahap ini berfokus terhadap penghapusan indentasi di setiap baris source codes dan spasi tambahan yang tidak diperlukan. Tahap ini memiliki tujuan untuk menghapus data yang dapat merusak hasil dari string matching sehingga data mudah untuk diproses pada tahap string matching.



* + 1. Tokenize

Pada tahap ini, sistem mengganti seluruh *identifier* variabel, tipe, dan konstanta menjadi token, kecuali *language construct* seperti lambang operator. Penggantian ini membuat porsi kode dengan variabel yang berbeda nama dapat terdeteksi sebagai *clone*.



* 1. Membandingkan Token

Sistem yang akan dirancang akan mendeteksi *code clone* pada level fungsi. Satu fungsi yang telah ditemukan akan dibandingkan dengan satu fungsi lain untuk diperiksa *code* *pattern* menggunakan algoritma *string matching*. Kedua fungsi yang memiliki *code* *pattern* yang mirip dengan fungsi lain disebut *clone pair.*

Sistem yang akan dirancang memiliki dua model algoritma *string matching* yang digunakan untuk menjadi eksperimen efisiensi mendeteksi *code clones*. Model pertama menggunakan algoritma *Knuth Morris Pratt pattern searching*. Sistem akan mencatat waktu yang dibutuhkan untuk menjalankan algoritma *string matching* yang dipakai. Hasil *clone pair* dan waktu berjalan setiap model algoritma merupakan *output* dari tahap ini.

* 1. Format Output

Pada tahap ini, sistem mengeluarkan seluruh informasi fungsi yang ada pada *source codes*, *clone pairs*, dan waktu yang diperlukan untuk mendeteksi *code clones*. Penamaan function yang dikeluarkan sistem ditambah dengan nama file sebagi *prefix* yang dipisah oleh tanda “.”, misalkan fungsi A terdapat pada file bernama X sehingga penamaan fungsi menjadi “X.A”.

*Output* pada tahap ini berupa file berformat excel yang memiliki dua kolom, *function pairs* dan *clone pairs*.Kolom *function pairs* berisi pasangan dua fungsi yang ada pada *source codes* dan kolom *clone pairs* berisi nilai “yes” atau “no” tergantung kepada pasangan dua fungsi kolom *function pairs*. Kolom *clone pairs* bernilai “yes” jika pasangan dua fungsi merupakan *code clones* dan bernilai “No” jika pasangan dua fungsi bukan merupakan *code clones*. Informasi waktu yang diperlukan untuk mendeteksi *code clones* terdapat pada baris terakhir pada file excel yang dipisahkan dengan baris dengan nilai *function pairs* “End”. Tabel 3‑1 adalah contoh *output* dari tahap ini.

|  |  |
| --- | --- |
| Function Pairs | Clone Pairs |
| File A.Function A-File A.Function B | Yes |
| File A.Function A-File A.Function C | No |
| End |  |
| Time | 59 (minutes) |

Evaluasi Hasil

Hasil dari tahap format output akan dibandingkan dengan hasil dari penelitian “A Code Clone Oracle” oleh Krutz dan Le [3]. Sistem akan menghitung jumlah prediksi clone pairs yang benar sesuai hasil penelitian Krutz dan Le dan menghitung akurasi dengam menggunakan Equation 3‑1 Perhitungan Akurasi Sistem Pendeteksi Code Clones. Untuk mengevaluasi performansi waktu, nilai waktu yang didapatkan pada format output akan dibandingkan dengan berapa Line Of Codes dari source codes, sehingga akan didapatkan nilai waktu yang diperlukan untuk menjalankan source codes dengan LOC tertentu.

**3. Results and Discussion**

**4. Conclusion**

Provide a statement that what is expected as stated in the Introduction section, and obtained in Results and Discussion section. Moreover, it can also be added the prospect of the development of research results and application prospects of further studies into the next (based on result and discussion).

**Notation**

The example of notation can be described with the following description:

n : the number of data

Mi : the median value of i-th class.

: the average value of data.

Fi : Frequency of i-th data.

Acknowledgement

Thank you to the Internet of Things Studio, Telkom University which has become a place for researchers to develop this journal research. Hopefully, this research can make a major contribution to the advancement of technology in Indonesia.

**References**

The minimal number of references used in manuscript must be 25 references using APA Style. The main references are reputed international journals or proceeding and doi hyperlink must be provided. All references should be to the most pertinent and up-to-date sources. In reference and citation are mandatory using reference apps such as Mendeley. Please use the reference writing procedure provided in this guide:

References

1. M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, “Internet of Things (IoT): Research, Simulators, and Testbeds,” *IEEE Internet of Things Journal*., Vol. 5, No. 3, Pp. 1637–1647, 2018. <https://doi.org/10.1109/JIOT.2017.2786639>