

BAB 6

KESIMPULAN DAN SARAN

Pada bab ini, akan dijelaskan kesimpulan dari skripsi ini dan saran untuk penelitian selanjutnya.

6.1 Kesimpulan

Berikut adalah kesimpulan dari penelitian ini:

- Algoritma genetik berhasil diimplementasi untuk melakukan pembangkitan kasus tes.
- Pengujian set kasus tes untuk mengetahui apakah set kasus tes sudah cukup baik untuk melakukan tes sudah dilakukan.
- Walaupun diotomasi, terdapat beberapa hal yang harus dilakukan manual. Seperti pembangunan kelas temp dan pencarian variabel *genetic algorithm* yang tepat untuk mencapai hasil maksimal/optimal.
- Terdapat banyak kelemahan pada kakas *TRGeneration* yang membatasi perangkat lunak ini. Keterbatasan ini juga salah satu alasan mengapa pembangunan kelas temp tidak diotomasi.
- Perangkat lunak untuk membangun set kasus tes secara otomatis dengan *genetic approach* berhasil dibangun dengan catatan harus membangun *filetemp*.
- Dalam kode program masukan, tidak semua tipe data dalam statement inialisasi dapat terdeteksi oleh kakas *TRGeneration*.

6.2 Saran

Saran untuk penelitian selanjutnya:

- Pembangunan *control-flow diagram* dan *test-requirement* dapat dilakukan sendiri menggunakan kakas *parsing*, salah satunya adalah *ANTLR4*, kakas untuk membangun *parser*. Dengan hal ini dilakukan, pembangunan kelas temp dapat dilakukan secara otomatis.
- Jika dilakukan *parsing* menggunakan kakas sendiri(seperti poin pertama), akan lebih mudah dalam memperbanyak jenis parameter kode program masukan yang dapat diproses perangkat lunak.

DAFTAR REFERENSI

- [1] A. S. Rosa and M. Shalahuddin., *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika, Bandung, 2nd ed., 2013.
- [2] R. L. Timothy C. Lethbridge, *Object-Oriented Software Engineering: Practical Software Engineering Using UML and Java*. Mcgraw-Hill College, 2nd ed., 2001.
- [3] R. Storn and K. Price., *Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces.* ". Journal of global optimization, 1997.
- [4] <https://github.com/evplatt/TRGeneration>.
- [5] P. Roger S. Pressman, *Test-data generation using genetic algorithms. Software Testing Verification and Reliability*. 1999.
- [6] A. P. Engelbrecht, "Computational intelligence: An introduction," *Software Quality Journal* 11.2, 2003.
- [7] M. J. H. Pargas, Roy P. and R. R. Peck., "Test-data generation using genetic algorithms.," *Software Testing Verification and Reliability* 9.4, 1999.
- [8] [http://www.programming-algorithms.net/article/45658/Fibonacci series](http://www.programming-algorithms.net/article/45658/Fibonacci%20series).
- [9] <http://www.programming-algorithms.net/article/39957/Factorial>.
- [10] [http://www.programming-algorithms.net/article/43434/Greatest-common divisor](http://www.programming-algorithms.net/article/43434/Greatest-common%20divisor).
- [11] [http://www.programming-algorithms.net/article/42865/Least-common multiple](http://www.programming-algorithms.net/article/42865/Least-common%20multiple).
- [12] [https://codereview.stackexchange.com/questions/18463/determining-triangle-type-from-three-integer inputs](https://codereview.stackexchange.com/questions/18463/determining-triangle-type-from-three-integer-inputs).
- [13] [http://www.programming-algorithms.net/article/44972/Quadratic equation](http://www.programming-algorithms.net/article/44972/Quadratic%20equation).