

Pemanfaatan Prusti untuk Memverifikasi Bahasa Pemrograman Rust = Utilization of Prusti for Verifying the Rust Programming Language

Al Ghifari Enerza Sentanu, author

Deskripsi Lengkap: <https://lib.ui.ac.id/detail?id=9999920563990&lokasi=lokal>

Abstrak

Rust adalah bahasa pemrograman sistem yang dikenal akan fitur keamanannya seperti ownership dan borrowing yang menjamin keamanan pada memori. Namun tidak semua keamanan dijamin oleh compiler Rust. Salah satunya karena adanya keyword unsafe yang tidak dicek oleh compiler. Prusti adalah alat verifikasi formal untuk bahasa Rust. Penelitian ini bertujuan untuk mengevaluasi seberapa akurat Prusti dalam memverifikasi program Rust. Eksperimen dilakukan dengan melakukan uji coba pada topik seperti conditional, operasi matematika, recursion, loop, indexing, borrowing dan fungsi unsafe. Program tersebut diverifikasi menggunakan Prusti lalu keluaran dari verifikasi dibandingkan dengan hasil yang diharapkan. Hasil penelitian menunjukkan Prusti akurat dalam memverifikasi program sederhana, namun memiliki keterbatasan untuk menangani fitur Rust yang belum didukung oleh Prusti seperti keyword unsafe. Untuk penelitian lanjutan, disarankan memperluas cakupan eksperimen ke fitur Prusti lainnya. Meskipun demikian, penelitian ini telah menunjukkan potensi Prusti dalam meningkatkan jaminan keamanan program Rust.

.....Rust is a systems programming language known for its safety features such as ownership and borrowing that guarantee memory safety. However, not all security aspects are guaranteed by the Rust compiler. The presence of the unsafe keyword allows code to bypass compiler checks. Prusti is a formal verification tool for the Rust language. This study aims to evaluate the accuracy of Prusti in verifying Rust programs. Experiments were conducted by evaluating topics such as conditionals, mathematical operations, recursion, loops, indexing, mutable borrows, and unsafe functions. The programs were verified using Prusti, and the outputs were compared with the expected results. The results show that Prusti is quite accurate in verifying simple programs but has limitations regarding Rust features that are not yet supported by Prusti, such as unsafe code. For further research, it is suggested to expand the scope of experiments to other Prusti features. Nevertheless, this study has demonstrated Prusti's potential in enhancing the security assurances of Rust programs.