

INTISARI

PENGARUH OPERASI *FINITE AUTOMATA* PADA *GRAMMAR SPECIFICATION* JAVACC TERHADAP PERFORMA *PARSER* YANG DIHASILKAN

Oleh

Baihaki Dwi Wahyu Kuncara

09/289059/PA/12901

Salah satu faktor yang mempengaruhi kecepatan pembacaan *parser* adalah proses *lookahead*. Proses ini digunakan untuk memilih satu dari beberapa aturan produksi yang mungkin untuk dipilih pada proses pembacaan. Proses ini dilakukan dengan melakukan pembacaan tambahan pada masukan. Dikarenakan adanya proses pembacaan tambahan, kecepatan pembacaan *parser* menjadi melambat.

Salah satu bentuk *grammar* yang dapat digunakan pada *grammar specification* JavaCC adalah *right-linear grammar*. Bentuk ini bisa diubah ke bentuk *finite automata* secara ekuivalen. Bentuk *finite automata* sendiri dapat diubah dari *non-deterministic finite automata* ke bentuk *deterministic finite automata*. Melalui pengubahan ini, maka tidak ada lagi ambiguitas untuk memilih transisi yang tepat, sehingga jika pengubahan ini digunakan pada *grammar* dari *parser*, *parser* tidak memerlukan proses *lookahead*.

Pada penelitian ini, dicoba untuk mengenakan operasi *finite automata*, yaitu pengubahan bentuk *non-deterministic finite automata* ke bentuk *deterministic finite automata* serta minimalisasi jumlah *state* dari *deterministic finite automata*, pada *grammar specification*. Hasil pengenaan dibandingkan dengan *parser* asli untuk mengetahui pengaruhnya terhadap kecepatan pembacaan. Dari perbandingan, ditunjukkan bahwa *parser* yang dikenakan operasi *finite automata* lebih lambat dibandingkan *parser* asli. Hal ini disebabkan karena lonjakan pada jumlah variabel dan aturan produksi serta pemecahan *string*. Hal ini mengakibatkan tambahan waktu yang lebih tinggi dibandingkan tambahan waktu dari *lookahead* yang dihilangkan.

Kata kunci: kecepatan, *parser*, JavaCC, *finite automata*, *lookahead*

ABSTRACT

THE EFFECT OF FINITE AUTOMATA OPERATIONS ON JAVACC'S GRAMMAR SPECIFICATION ON THE GENERATED PARSER'S PERFORMANCE

Oleh

Baihaki Dwi Wahyu Kuncara

09/289059/PA/12901

One of the factors that affect the speed of parser readings is the lookahead process. This process is used to select one of several production rules that may be selected in the reading process. This process is done by performing additional readings on the input. Due to an additional reading process, the parser reading speed slows down.

One form of grammar that can be used in JavaCC's grammar specification is the right-linear grammar. This form can be converted to an equivalent form of finite automata. The shape of the finite automata itself can be changed from non-deterministic finite automata to the deterministic finite automata. Through this conversion, there is no ambiguity to choose the right transition, so if this conversion is used on the grammar of the parser, the parser does not require a lookahead process.

In this study, it is attempted to impose finite automata operations, namely the non-deterministic finite automata conversion to the deterministic finite automata and the minimization of the number of states of deterministic finite automata, to the grammar specification. The imposition results are compared with the original parser to determine its effect on the readout speed. From the comparison, it is shown that the parser subjected to finite automata operation is slower than the original parser. This is due to a spike in the number of variables and production rules as well as breaking the string. This results in an additional time higher than the additional time of the removed lookahead.

Keywords: performance, parser, JavaCC, finite automata, lookahead