

PERBANDINGAN KEMAMPUAN REGRESSION TESTING TOOL PADA REGRESSION TEST SELECTION: STARTS DAN EKSTAZI

Asri Maspupah¹, Akhmad Bakhrun²

^{1,2}Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung
Email: ¹maspupah@polban.ac.id, ²abakhrun@polban.ac.id

Abstrak

Regression testing sebagai aktivitas esensial pada pengembangan *software* yang mengalami perubahan *requirement*. Pada pelaksanaannya, *regression testing* memerlukan waktu yang tidak sedikit sehingga dibutuhkan strategi yang optimal. Salah satu pendekatan yang dapat digunakan untuk mempercepat waktu pelaksanaan yaitu pendekatan *Regression Test Selection (RTS)*. Saat ini, para praktisi maupun akademisi sudah mulai berfikir untuk mengembangkan tools dalam mengoptimalkan proses pelaksanaan *regression testing*. Diantaranya, STARTS dan Ekstazi merupakan *regression testing tools* yang paling populer di kalangan akademisi dalam menjalankan algoritma pemilihan *test case*. Artikel ini membahas perbandingan kemampuan fitur STARTS dan Ekstazi dengan menggunakan evaluasi parameter *feature*. Kedua tool tersebut diuji coba dengan data input berupa System Under Test (SUT) dan test case yang sama. Parameter yang digunakan dalam perbandingan tool adalah *platform technology*, pemilihan *test case*, *functionality*, *usability* dan *performance efficiency*, kelebihan serta kekurangan tool. Hasil uji coba menunjukkan perbedaan dan persamaan fitur STARTS dan Ekstazi, sehingga dapat digunakan oleh praktisi untuk memanfaatkan tool pada pelaksanaan *regression testing* yang sesuai dengan kebutuhannya. Selain itu, hasil eksperimen menunjukkan bahwa penggunaan Ekstazi lebih tepat dalam memilih *test case* yang penting dan lebih efisien, jika dibandingkan dengan STARTS dan pengujian *regressi* dengan *retest all*.

Kata Kunci: perbandingan kemampuan tool, regression testing, regression test selection, STARTS, Ekstazi

Abstract

Regression testing as an essential activity in software development that has changed requirements. In practice, regression testing requires a lot of time so that an optimal strategy is needed. One approach that can be used to speed up execution time is the Regression Test Selection (RTS) approach. Currently, practitioners and academics have started to think about developing tools to optimize the process of implementing regression testing. Among them, STARTS and Ekstazi are the most popular regression testing tools among academics in running test case selection algorithms. This article discusses the comparison of the capabilities of the STARTS and Ekstazi features by using feature parameter evaluation. Both tools were tested with the same input data in the form of System Under Test (SUT) and test cases. The parameters used in the tool comparisons are platform technology, test case selection, functionality, usability and performance efficiency, the advantages, and disadvantages of the tool. The results of the trial show the differences and similarities between the features of STARTS and Ekstazi, so that it can be used by practitioners to take advantage of tools in the implementation of regression testing that suit their needs. In addition, experimental results show that the use of Ekstazi is more precise in sorting out important test cases and is more efficient, when compared to STARTS and regression testing with retest all.

Keywords: study comparative tools, regression testing, regression test, STARTS, Ekstazi

I. PENDAHULUAN

Regression testing merupakan salah satu aktivitas pengujian *software* yang esensial digunakan pada metode pengembangan *software* dengan siklus perubahan secara berskala pada agile development (Faizullah and Almutairi, 2018). *Regression testing* digunakan untuk menemukan *defect* atau *error* pada bagian *software* yang sudah dipastikan benar sebelumnya sebagai akibat dari *software* yang telah dimodifikasi (Simanca *et al.*, 2019) (Rosero, Gómez and Rodríguez, 2017) (Ali *et al.*, 2019). Proses pelaksanaan *regression testing*, yaitu menjalankan kembali *test case* yang telah dijalankan untuk memeriksa *defect* baru atau *defect* yang telah diperbaiki muncul kembali (Rosero, Gómez and Rodríguez, 2017).

Metode yang paling umum digunakan oleh praktisi industri dalam melaksanakan *regression testing* adalah pemilihan *test case* berdasarkan *experience based* dari tester. Namun strategi ini sulit dilakukan untuk memilih *test case* yang relevan dalam menguji kode program yang mengalami perubahan. Sementara pada strategi lain, pemilihan *test case* dengan *retest all*, yaitu penggunaan kembali semua *test case* untuk menjamin kebenaran *software* akibat adanya perubahan kode program. Meskipun dengan pendekatan tersebut proses pengujian regresi memakan banyak waktu dan sumber daya (Engström and Runeson, 2010)(Ansari *et al.*, 2016).

Beberapa peneliti mulai tertarik membahas metoda pelaksanaan *regression testing tool* untuk meminimalkan waktu pengujian. Salah satunya dengan memilih *test case* yang penting untuk dijalankan pada pelaksanaan pengujian regresi.

Regression Test Selection (RTS) merupakan salah satu metode untuk mempercepat waktu pelaksanaan pengujian dengan memilih *test case* yang penting untuk dijalankan berdasarkan kriteria tertentu (Gligoric, Eloussi and Marinov, 2015). Saat ini penelitian mengenai RTS sudah banyak yang mengusulkan beberapa algoritma dalam pemilihan *test case*. Namun, algoritma yang paling populer diusulkan oleh (Legunsen *et al.*, 2016) (Legunsen, Shi and Marinov, 2017) dan (Gligoric, Eloussi and Marinov, 2015), yaitu algoritma *static* dan *lightweight regression test selection*.

Dalam penelitiannya, Legunsen membuat STARTS dengan menggunakan algoritma *static regression test selection* untuk membuktikan algoritmanya dapat mengurangi waktu pelaksanaan pengujian. Penelitian Legunsen telah dirujuk sebanyak 29 kali oleh para peneliti untuk membuktikan hipotesanya dalam berbagai kasus.

Sementara, Gligoric membuat Ekstazi dengan menggunakan algoritma *lightweight regression test selection*. Tujuan penelitian Gligoric hampir sama dengan Legunsen untuk mencari algoritma RTS dalam mencari *test case* yang efisien dan efektif dalam menemukan *defect* karena adanya perubahan kode program. Penelitian Gligoric telah dirujuk sebanyak 39 kali oleh para peneliti.

Berdasarkan fenomena di atas, *regression testing tools* dengan metode RTS mampu mengurangi waktu melaksanakan pengujian pada pendekatan *retest all*. Dengan demikian, penelitian ini membahas perbandingan kemampuan *tools*, yaitu STARTS dan Ekstazi, untuk melaksanakan *regression testing* yang menggunakan metode RTS. Perbandingan *regression testing tools* bertujuan untuk melihat persamaan dan perbedaan fitur *tool* agar dapat digunakan oleh praktisi sesuai dengan kebutuhannya. Manfaat dari penelitian ini, para praktisi mendapatkan pengetahuan tentang kajian kedua *tools* tersebut berdasarkan kriteria yang telah ditetapkan. Secara umum kriteria tersebut digunakan untuk mengetahui instalasi *tools*, objektif *tools*, cara kerja *tools*, dan kinerja *tools*.

Dengan demikian, penelitian ini menekankan kepada *study comparative regression testing tools* dengan membandingkan kemampuan *tools* berdasarkan kriteria perbandingan. Pada penelitian sebelumnya, telah dilakukan *study comparative testing tools*. Fungsi *study comparative testing tools* dapat membantu pengguna untuk memilih *software testing tools* yang sesuai berdasarkan kebutuhan (Gamido and Gamido, 2019). Salah satu peneliti yaitu Gamido pada tahun 2019 melakukan *study comparative automation testing tools*. Pada penelitiannya, Gamido membandingkan kemampuan *tools* yang umum digunakan dalam *automation testing* sesuai dengan tujuan *software testing* secara umum, yaitu menemukan *defect* agar *software* sesuai dengan *requirement*.

Sistematika penulisan artikel ilmiah ini disusun sebagai berikut: Sesi 1 membahas latar belakang dan tujuan penelitian. Sesi 2 membahas metode yang digunakan pada penelitian ini. Sesi 3 membahas diskusi hasil penelitian. Terakhir sesi 4 membahas kesimpulan.

II. METODE

Metode penelitian sebagian besar mengadopsi dari tahapan metode yang dipaparkan oleh (Gamido and Gamido, 2019) pada artikel ilmiah yang berjudul “Comparative Review of the Features of Automated Software Testing Tools”. Penelitian dimulai dengan mengidentifikasi *tools* yang akan dikaji. Kemudian menentukan kriteria pembandingan sesuai dengan

objektif penelitian. Tahap ketiga melakukan eksperimen dengan membandingkan tools berdasarkan kriteria pembandingan. Terakhir melakukan evaluasi dan pemberian kesimpulan terhadap studi komparatif tools.

Perbedaan penelitian terdahulu dengan penelitian ini terletak pada aspek objektif penelitian dan kriteria pembandingnya. Objektif penelitian (Gamido and Gamido, 2019) adalah membandingkan automation testing tools, sedangkan objektif fokus kepada pembahasan tools yang dimanfaatkan untuk mendukung proses *regression testing*. Disisi lain, kriteria pembandingan pada penelitian terdahulu menggunakan kriteria yang umum digunakan pada *automation testing*, seperti *cross browser*, *record playback*, *script language*, *data driven*, dan *report generation*, sedangkan kriteria pembandingan evaluasi tools penelitian ini menekankan kepada kinerja tools terhadap performa *regression testing*.

Penjelasan metode yang digunakan pada penelitian ini dipaparkan pada sub bab di bawah ini.

2.1. Regression Testing Tools

Regression testing tools yang dipilih sebagai kajian penelitian pada teknik *regression test selection* adalah STARTS dan Ekstazi.

- (a) STARTS adalah *static regression test selection* untuk menguji *System Under Test* (SUT) yang dikembangkan pada *base project java maven*. STARTS melakukan analisis statis pada dependency keterkaitan antar *class* untuk menemukan *test case* yang terkena dampak perubahan kode program (Legunsen, Shi and Marinov, 2017).
- (b) Ekstazi adalah *java library* yang ringan untuk menguji SUT yang dikembangkan pada *base project java maven*. Ekstazi melakukan pembobotan penskalaan untuk *regression test selection* berdasarkan file dependency (Gligoric, Eloussi and Marinov, 2015).

STARTS dan Ekstazi merupakan *open source* untuk melakukan pengujian regresi yang didapatkan dari penyedia tool. STARTS dapat diunduh pada link:

<https://github.com/TestingResearchIllinois/starts>.

Sementara Ekstazi dapat diunduh pada link: <http://www.ekstazi.org>.

STARTS dan Ekstazi dibuat oleh para akademisi untuk kebutuhan penelitian dalam bidang kajian *regression testing tools*. Akademisi yang berkontribusi dalam pembuatan STARTS adalah Owolabi Legunsen, August Shi, dan Darko Marinov. Sementara akademisi adalah berkontribusi dalam pembuatan Ekstazi adalah Milos Gligoric, Lamyaa Eloussi, and Darko Marinov.

2.2. Evaluasi Parameter Feature

Agar diketahui perbedaan dan persamaan fitur dari STARTS dan Ekstazi maka diperlukan parameter pembandingan sebagai indikator perbandingan tool pada saat uji coba tool (Gamido and Gamido, 2019). Parameter ini disebut sebagai parameter *feature* karena berfungsi sebagai dasar perbandingan untuk melihat kemampuan dan kinerja tool pada saat melakukan *regression testing*.

Teridentifikasi pada beberapa penelitian bahwa untuk melakukan perbandingan dapat mempertimbangkan beberapa poin utama sebagai parameter pembandingan. Parameter pembandingan yang teridentifikasi adalah *support to platforms and technology*, *functionality*, *usability*, *performance efficiency* dan kelebihan serta kekurangan tools (Sandin, Yassin and Mohamad, 2016) (Satheesh and Singh, 2017) (Kannan, M; Lokeshwari, 2017) (Rana and Yousuf, 2019) (Gamido and Gamido, 2019). Dengan demikian, evaluasi *parameter feature* yang digunakan pada penelitian ini ditunjukkan pada Tabel 1. Evaluasi Parameter Feature Tabel 1.

Tabel 1. Evaluasi Parameter Feature

Parameter Evaluation	Deskripsi	Referensi
<i>Cross Platform</i>	Sistem operasi yang mendukung. Aspek yang dikaji: platform	(Gamido and Gamido, 2019) (Satheesh and Singh, 2017) (Kannan, M; Lokeshwari, 2017)
<i>Functionality</i>	kemampuan tools untuk melakukan <i>regression testing</i> . Aspek yang dikaji: Ability, kriteria pemilihan <i>test case</i> , teknik pemilihan <i>test case</i> ; <i>run tool</i> , dan <i>output tools</i>	(Satheesh and Singh, 2017) (Kannan, M; Lokeshwari, 2017) (Sandin, Yassin and Mohamad, 2016)
<i>Usability</i>	Kemudahan dalam memahami dan mempelajari tools. Aspek yang dikaji: Pemahaman dokumentasi, mudah digunakan, dan mudah untuk diinstalasi.	(Gamido and Gamido, 2019) (Satheesh and Singh, 2017) (Sandin, Yassin and Mohamad, 2016)

Parameter Evaluation	Deskripsi	Referensi
<i>Performance efficiency</i>	Tingkat kinerja pada beberapa kondisi. Aspek yang dikaji: <i>Run time execution</i>	(Gamido and Gamido, 2019) (Satheesh and Singh, 2017)
<i>Other</i>	Temuan pada saat dilakukan uji coba <i>tools</i> . Aspek yang dikaji: Kelebihan dan kekurangan	(Gamido and Gamido, 2019) (Rana and Yousuf, 2019)

2.3. Perbandingan Kemampuan Tool

Perbandingan kemampuan *tool* digunakan untuk menunjukkan perbedaan dan persamaan STARTS dan Ekstazi. Proses perbandingan *tool* menggunakan metode *exploratory study* berdasarkan uji coba eksperimen (Edgar and Cyber, 2017). Setiap *tool* diuji coba dengan menjalankannya untuk mengetahui *defect* pada skema pelaksanaan proses *regression testing*. Skema pengujian pada *regression testing* melibatkan dua kali pengujian untuk mengetahui *defect* pada pengujian ulang sebagai dampak dari perubahan *requirement* (Ali et al., 2019). Dengan demikian, eksperimen perbandingan performa kedua *tools* melalui pengamatan proses *regression testing* pada pengujian SUT yang kedua. Pengamatan dilakukan berdasarkan evaluasi *parameter feature*.

Proses *regression testing* dilakukan terhadap sebuah kasus uji coba pada SUT yang telah dimodifikasi. Skema uji coba mengadopsi pengujian regresi yang melibatkan dua kali proses pengujian.

Pengujian pertama, pengujian SUT setelah implementasi kode program selesai. Pengujian ini bertujuan untuk mengetahui *defect* sebagai akibat kesalahan kode program yang tidak sesuai dengan *requirement*. Setelah pengujian pertama dilakukan, dihasilkan *defect* kemudian diperbaiki sehingga sesuai dengan *requirement*. Pengujian pertama memastikan bahwa SUT telah diimplementasikan dengan benar sesuai dengan *requirement*.

Pengujian kedua, pengujian SUT setelah implementasi perubahan kode program selesai. Pengujian ini bertujuan untuk mengetahui *defect* pada bagian kode program yang tidak terkena dampak perubahan. Jika terjadi *defect*, hal ini disebabkan oleh kesalahan perubahan kode program. Pengujian kedua memastikan bahwa

perubahan kode program tidak mengakibatkan *defect* pada kode program yang tidak terkena dampak perubahan.

Uji coba pelaksanaan *regression testing* pada STARTS dan Ekstazi menggunakan data SUT dan test case yang sama. Karakteristik SUT dan *test case* ditunjukkan pada Tabel 2.

Tabel 2. Karakteristik Data Input Eksperimen

Jenis Data Masukan	Deskripsi	Data Masukan
<i>System Under Test (SUT)</i>	Kode program yang akan diuji.	JodaTime
<i>Test case</i>	Sekumpulan data skenario <i>test case</i> yang dibuat dalam bentuk program pada <i>level unit testing</i> .	Script test joda time

JodaTime sebagai SUT eksperimen, dimana *source code*-nya dapat diunduh pada link: <https://github.com/JodaOrg/joda-time>. Sementara dokumentasinya dapat dilihat pada link: <https://www.joda.org/joda-time/>.

Selanjutnya dilakukan pengamatan terhadap proses, output dan temuan-temuan berdasarkan *parameter feature* pada saat uji coba eksperimen STARTS dan Ekstazi. Kemudian dilakukan proses analisa perbandingan untuk mengetahui persamaan dan perbedaan fitur dari STARTS dan Ekstazi.

III. HASIL DAN PEMBAHASAN

Hasil uji coba eksperimen berupa perbandingan *parameter feature* pada kemampuan *tool*.

3.1. Cross Platform

Cross platform adalah *parameter feature* yang membandingkan sistem operasi yang didukung oleh *tool*. Perbandingan parameter *cross platform* dijelaskan di bawah ini.

- (a) STARTS dijalankan di LinuxOS dan MacOS.
- (b) Ektazi dijalan di Windows dan LinuxOS.

3.2. Functionality

Functionality adalah *parameter feature* yang membandingkan kemampuan *tool* dalam menjalankan proses *regression testing*. Perbandingan *functionality* pada STARTS dan Ekstazi ditunjukkan pada Tabel 3.

Berdasarkan Tabel 3, perbandingan *functionality* pada STARTS dan Ekstazi menunjukkan adanya persamaan dan perbedaan. Persamaan terletak pada aspek *ability* dalam melakukan *regression testing*, kriteria pemilihan *test case*, bagaimana menjalankan *tool* dan keluaran dari *tool*. Sementara perbedaan terletak pada algoritma dalam memilih *test case*

yang sesuai ketika terjadi perubahan kode program, yaitu pada aspek teknik pemilihan *test case*.

Tabel 3. Perbandingan Parameter Functionality

<i>Parameter feature</i>	STARTS	Ekstazi
<i>ability</i>	Memilih <i>test case</i> yang penting untuk dijalankan ulang pada pengujian regresi. Kemudian menjalankan <i>test case</i> terpilih untuk melihat <i>defect</i> yang terkena dampak perubahan.	
kriteria pemilihan <i>test case</i>	Kriteria yang digunakan berdasarkan perubahan kode program sebagai akibat modifikasi fitur	
teknik pemilihan <i>test case</i>	Pembuatan <i>dependency graph</i> untuk mencari keterkaitan antar <i>class</i> . Selanjutnya melakukan perhitungan <i>checksum</i> pada setiap <i>test case</i> untuk memilih <i>test case</i> penting berdasarkan perubahan kode program (Legunsen <i>et al.</i> , 2016).	Pembuatan <i>dependency file</i> berdasarkan <i>invoke method</i> . Selanjutnya perhitungan fungsi <i>checksum</i> pada setiap <i>script test case</i> penting berdasarkan perubahan kode program (Gligoric, Eloussi and Marinov, 2015).
<i>run tool</i>	Pengaturan konfigurasi SUT pada pom.xml dengan menambahkan <i>maven plugin</i> . SUT yang akan diujikan haruslah <i>project java</i> dengan <i>base maven plugin</i> . Kemudian menjalankan <i>tool</i> melalui instruksi <i>command line</i> pada <i>path working folder</i> SUT.	
output tools	Menjalankan berdasarkan <i>test case</i> terpilih kemudian memberikan informasi berupa <i>log testing</i> . <i>Log testing</i> berisi jumlah <i>test case</i> yang di-running, jumlah <i>failure</i> , waktu eksekusi dan status <i>build</i> .	

STARTS dan Ekstazi memandang pemilihan *test case* berdasarkan perubahan kode program pada saat modifikasi fitur dengan melihat *dependency file* atau keterkaitan antar *class* pada SUT. Selanjutnya mencari *test case* penting yang relevan berdasarkan perubahan kode program dengan menggunakan *checksum*. Namun algoritma yang digunakan sebagai kriteria dasar untuk membuat *dependency* memiliki cara yang berbeda. STARTS membuat *dependency graph*, sedangkan Ekstazi membuat *dependency file* berdasarkan *invoke method*.

Pada saat pemilihan *test case* relevan dengan perubahan kode program, algoritma yang digunakan pada STARTS dan ekstazi menggunakan fungsi *checksum*. Fungsi *checksum* menghasilkan sebuah string yang mewakili isi kode program dan tipe tertentu. Nilai *checksum* akan berubah ketika terjadi

perubahan kode program (Gligoric, Eloussi and Marinov, 2015).

Nilai *checksum* menentukan apakah kode program yang sedang diuji mengalami perubahan. Perubahan diketahui jika terdapat perbedaan nilai *checksum* dengan nilai sebelumnya. Hasil fungsi *checksum* disimpan di dalam suatu file perhitungan. STARTS dan Ekstazi akan menggunakan nilai *checksum* untuk menentukan apakah *test case* perlu digunakan kembali pada pengujian regresi.

STARTS memanfaatkan nilai *checksum* untuk melakukan analisis statis dengan menggunakan *dependency graph*. Analisis statis tersebut bertujuan untuk mengetahui keterkaitan antar *class* pada *source code*. Selanjutnya dilakukan analisa pemilihan *test case* dengan menganalisis apakah *test case* dapat meraih suatu SUT yang mengalami perubahan kode program pada *dependency graph*. Apabila *test case* tersebut memiliki keterhubungan dengan SUT maka *test case* tersebut terpilih menjadi *test case* yang akan diujikan kembali. Kemudian STARTS menyimpan hasil perhitungan *checksum* dalam satu file yang berisi sekumpulan test (set of test) yang bergantung kepada suatu file tertentu (Legunsen, Shi and Marinov, 2017).

Sementara Ekstazi memanfaatkan nilai *checksum* untuk melakukan analisis sederhana. Perhitungan nilai *checksum* dilakukan pada setiap *test case*. Pada saat Ekstazi digunakan pada saat pertama kali, tidak ada informasi *dependence* untuk semua *class*, sehingga semua *class* dipilih untuk *retest all*. Pada akhir proses setiap *test case* memiliki nilai *checksum*. Pada pengujian kedua, ekstazi memeriksa apakah nilai *checksum* pada file *dependence* masih sama. Jika sama maka *test case* tersebut tidak dipilih (Gligoric, Eloussi and Marinov, 2015).

3.3. Usability

Usability adalah *parameter feature* yang membandingkan kemudahan dalam proses pembelajaran *tools*. Tingkat kemudahan pada ketiga *parameter usability* dikategorikan ke dalam mudah, sedang dan sulit. Penjelasan masing-masing kategori dijelaskan pada Tabel 4.

Setelah didapatkan hasil eksperimen perbandingan kemampuan tools dengan melihat parameter *usability*, selanjutnya adalah memetakan kemampuan *usability tools* ke dalam kategori tingkat kemudahan. Kategori tingkat kemudahan ini ditunjukkan pada Tabel 4. Berdasarkan analisa dari hasil eksperimen, maka didapatkan hasil perbandingan kemampuan STARTS dan Ekstazi yang ditinjau dari aspek parameter *usability* pada ditunjukkan pada Tabel 5.

Tabel 4. Penjelasan Tingkat Kemudahan

Kategori	Deskripsi
Mudah	<p>Aspek: Pemahaman Dokumentasi Tersedianya dokumentasi di jejaring internet baik yang disediakan oleh website penyedia tools atau para peneliti yang menjelaskan kinerja tools (Sandin, Yassin and Mohamad, 2016).</p>
	<p>Aspek: Mudah digunakan Sebuah prosedur memiliki langkah-langkah penggunaan yang mudah untuk diingat (Churm, 2017).</p>
	<p>Aspek Kemudahan Instalasi Pada saat instalasi memiliki pengaturan konfigurasi yang sederhana yaitu tidak banyak yang harus diatur (Satheesh and Singh, 2017).</p>
Sedang	<p>Aspek: Pemahaman Dokumentasi Tersedianya dokumentasi penjelasan tools, tetapi membutuhkan upaya agar kinerja tools dipahami (Sandin, Yassin and Mohamad, 2016).</p>
	<p>Aspek: Mudah digunakan Sebuah prosedur memiliki langkah-langkah penggunaan yang agak susah untuk diingat (Churm, 2017)</p>
	<p>Aspek Kemudahan Instalasi Jika ditinjau dari pernyataan statheesh maka yang termasuk kategori sedang yang tools yang memiliki pengaturan yang tidak terlalu sederhana, tetapi tidak terlalu kompleks.</p>
Sulit	<p>Aspek: Pemahaman Dokumentasi Tersedianya dokumentasi penjelasan tools, tetapi membutuhkan upaya lebih agar kinerja tools dipahami atau bahkan dokumentasi tidak tersedia (Sandin, Yassin and Mohamad, 2016)</p>
	<p>Aspek: Mudah digunakan Sebuah prosedur memiliki langkah-langkah penggunaan yang susah untuk diingat (Churm, 2017)</p>
	<p>Aspek Kemudahan Instalasi Pada saat instalasi memiliki pengaturan konfigurasi framework yang rumit yaitu banyak yang harus diatur (Satheesh and Singh, 2017)</p>

Tabel 5. Perbandingan Parameter Usability

Parameter feature	STARTS	Ekstazi
Pemahaman dokumentasi.	sedang	mudah
Mudah digunakan	mudah	mudah
Kemudahan instalasi	mudah	mudah

Tabel 5 menggambarkan perbandingan STARTS dan Ekstazi pada parameter *usability* berdasarkan aspek kajian tingkat kemudahan dalam pemahaman

dokumentasi, kemudahan penggunaan tools dan kemudahan instalasi tools. Aspek dokumentasi membahas mengenai tingkat kemudahan dalam memahami informasi terkait instalasi dan penggunaan tool yang disediakan tool.

Ditinjau dari aspek tingkat kemudahan pemahaman dokumentasi, STARTS termasuk ke dalam kategori sedang karena dokumentasi yang tersedia tidak lengkap sehingga untuk mendapatkan informasi instalasi dan penggunaan tool harus mencari sumber lain. Sementara Ekstazi sudah banyak tersedia dokumentasi yang menjelaskan instalasi dan penggunaan tools dengan bahasa yang mudah dimengerti.

Sementara ditinjau dari aspek kemudahan dalam penggunaan tool, STARTS dan Ekstazi termasuk ke dalam kategori mudah karena menggunakan instruksi yang sederhana melalui command line. STARTS dijalankan dengan instruksi `mvn starts:starts`, sedangkan ekstazi dijalankan dengan instruksi `ekstazi:ekstazi` pada path working folder SUT.

Disisi lain, jika ditinjau dari aspek kemudahan instalasi *tools*, STARTS dan Ekstazi termasuk ke dalam kategori mudah karena cukup melakukan pengaturan konfigurasi penambahan *library tools* pada SUT melalui `pom.xml` secara sederhana. Konfigurasi pada tahap instalasi *tool* bertujuan untuk menambahkan plugin STARTS atau Ekstazi di dalam *working folder* SUT. Pengaturan konfigurasi STARTS ditunjukkan pada Tabel 6, sedangkan konfigurasi Ekstazi ditunjukkan pada Tabel 7.

Tabel 6. Konfigurasi Penambahan Plugin STARTS

File	pom.xml
Sintaksis	
<pre><build> <plugins> <plugin> <groupId>edu.illinois</groupId> <artifactId>starts-maven- plugin</artifactId> <version>1.3</version> </plugin> </plugins> </build></pre>	

Tabel 7. Konfigurasi Penambahan Plugin Ekstazi

File	pom.xml
Sintaksis	
<pre><build> <plugins> <plugin> <groupId>edu.illinois</groupId> <artifactId>starts-maven- plugin</artifactId> <version>1.3</version> </plugin> </plugins> </build></pre>	

3.4. Performance Efficiency

Performance *efficiency* adalah *parameter feature* yang membandingkan tingkat kinerja dalam melakukan eksekusi *regression testing*. *Parameter feature performance efficiency* yang digunakan pada penelitian ini berdasarkan kecepatan waktu eksekusi. Uji coba dilakukan dengan mencatat waktu pada saat menjalankan STARTS, Ekstazi dan pengujian dengan pendekatan *retest all*. Perbandingan kecepatan pengujian regresi ditunjukkan pada Tabel 8.

Tabel 8. Perbandingan *Performance Efficiency*

<i>Parameter Feature</i>	<i>Run Time Execution</i>
STARTS	50 detik
Ekstazi	15.238 detik
<i>Retest All</i>	58.972 detik

Tabel 8 menunjukkan bahwa Ekstazi memiliki performa yang lebih baik dibandingkan dengan STARTS karena hasil perhitungan pemilihan test case dari metode yang digunakan lebih akurat. Disisi lain, jika dibandingkan dengan waktu yang dibutuhkan ketika pengujian regresi pada *retest all*, performa STARTS tidak terlalu signifikan, yaitu pengurangan sebesar 15% sebanyak 8.972 detik dari total waktu *retest all* (58 detik). Sementara, Ekstazi mampu mengurangi waktu pelaksanaan sebesar 74% sebanyak 43.734 detik dari total waktu *retest all* (58 detik). Dengan kata lain, pelaksanaan *regression testing* oleh Ekstazi lebih cepat 30% dibandingkan dengan STARTS. Hal ini menunjukkan bahwa proses pemilihan *test case* pada STARTS tidak jauh lebih baik karena adanya kemungkinan *test case* tetap dijalankan meskipun tidak terkena dampak perubahan kode program.

3.5. Other

Parameter feature yang membandingkan temuan-temuan pada saat uji coba eksperimen yang ditinjau dari tiga aspek yaitu kekurangan dan kelebihan tool. Penjelasan perbandingan *parameter feature other* dipaparkan di bawah ini.

a. Kelebihan *tool*

- 1) STARTS memiliki kemudahan pada saat pengaturan konfigurasi dengan SUT dan kemudahan pada saat digunakan. Hal ini selaras dengan hasil analisa komparatif yang ditinjau dari *parameter usability*.
- 2) Ekstazi memiliki kelebihan dalam pengecekan yang lebih akurat. Hal ini karena semua *method* dievaluasi dan mengidentifikasi kemungkinan yang terjadi

selama pengujian dijalankan. Misalnya, *defect* yang ditemukan. Pengecekan dengan tingkat akurasi ini akan memudahkan untuk mengetahui sisi mana yang terdapat *defect* atau di metode mana, baris kode, dan berapa banyak *defect* yang ditemukan (Legunsen *et al.*, 2016) (Legunsen, Shi and Marinov, 2017).

b. Kekurangan *tool*

- 1) STARTS memiliki kekurangan sbb:

Algoritma pemilihan *test case* yang digunakan berdasarkan perubahan kode program, terdapat perhitungan presisi yang tidak akurat. Hal ini menyebabkan gagalnya proses pemilihan *test case* yang penting sehingga besar kemungkinan tidak terjadi pengurangan *test case* (Legunsen, Shi and Marinov, 2017).

- 2) Sementara Ekstazi memiliki kekurangan sbb:

Pada saat pelaksanaan *regression testing*, Ekstazi sangat berpengaruh terhadap versi JUnit yang diintegrasikan. Pada saat diintegrasikan dengan JUnit versi 3, Ekstazi hanya mendukung pembuatan *dependency file* pada level *method* (bukan *class*). Sementara jika diintegrasikan dengan JUnit versi 4, Ekstazi hanya mendukung pembuatan *dependency file* pada level *class* (bukan *method*) (Gligoric, Eloussi and Marinov, 2015). Oleh karena adanya perbedaan versi *dependency file* yang tergantung dari versi JUnit yang digunakan, menurut penulis akan berdampak pada kebutuhan *regression testing*, yaitu jika ingin mengecek berdasarkan level *method* atau level *class* harus berganti versi JUnit.

IV. PENUTUP

Kesimpulan

Perbandingan kemampuan *regression test selection* pada kedua tools menunjukkan perbedaan performa yang ditinjau dari *parameter feature* yang telah ditentukan. Perbedaan tersebut terletak kepada *environment tools*, teknik pemilihan *test case*, tingkat keakuratan, kemudahan mendapatkan dokumentasi penggunaan *tools*, *time execution* dan instalasi *tools*. Perbandingan tersebut dapat pula menunjukkan persamaan, yaitu terletak pada *ability tools*, *output tools*, cara melakukan *running tools*, kriteria pemilihan *test case*, dan kemudahan dalam instalasi dan penggunaan *tools*.

Selain itu, perbandingan kemampuan *tools* pada proses pemilahan *test case* menunjukkan tingkat keakuratan *tools* dalam menjalankan *regression testing*. Analisa hasil eksperimen menunjukkan bahwa metoda *regression testing* dengan RTS mampu mengurangi waktu pelaksanaan *regression testing* pada metode *retest all*. Ekstazi mampu mengurangi waktu pelaksanaan pengujian sebanyak 74% dari *retest all*. Disisi lain, pemilihan *test case* bagian kode program yang harus diuji kembali sebagai dampak dari perubahan kode pada Ekstazi lebih cepat 30% dan lebih akurat dibandingkan dengan STARTS. Dengan demikian, Ekstazi merupakan pilihan yang tepat untuk RTS dengan tingkat keakuratan dan kecepatan eksekusi yang baik.

Pada akhir penelitian disimpulkan bahwa dari *parameter feature*, yaitu *cross platform, functionality, usability, performance efficiency*, kelebihan dan kekurangan *tools*, dapat digunakan untuk melihat perbedaan & persamaan fitur *tools* sehingga informasinya bermanfaat kepada praktisi dalam rangka memanfaatkan tool untuk pelaksanaan *regression testing* yang disesuaikan dengan kebutuhannya.

Saran

Berdasarkan temuan hasil penelitian ini, pengembangan penelitian berikutnya dapat dengan menambahkan *parameter feature* untuk mengevaluasi *regression testing tool*. Selain itu, penambahan *tools* yang dibandingkan, selain STARTS dan Ekstazi, agar dapat menambahkan variasi *tools* yang dapat dimanfaatkan pada *regression testing*.

V. DAFTAR PUSTAKA

- Ali, N. bin et al. (2019) On the search for industry-relevant regression testing research, *Empirical Software Engineering*. doi: 10.1007/s10664-018-9670-1.
- Ansari, A. et al. (2016) ‘Optimized Regression Test Using Test Case Prioritization’, *Procedia Computer Science*, 79, pp. 152–160. doi: 10.1016/j.procs.2016.03.020.
- Churm, T. (2017) An Introduction To Website Usability Testing, Usability Geek, [online]. Available at: <https://usabilitygeek.com/an-introduction-to-website-usability-testing/> (Accessed: 28 November 2020).
- Edgar, T. W. and Cyber, S. (2017) ‘About the Authors’, *Research Methods for Cyber Security*, p. xv. doi: 10.1016/b978-0-12-805349-2.00020-0.
- Engström, E. and Runeson, P. (2010) ‘A qualitative survey of regression testing practices’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6156 LNCS(June), pp. 3–16. doi: 10.1007/978-3-642-13792-1_3.
- Faizullah, S. and Almutairi, S. (2018) ‘Considerations for Regression Testing Process in Agile Development Environments’, *International Journal of Advanced Research in Computer Science and Software Engineering*, 8(1), p. 153. doi: 10.23956/ijarcsse.v8i1.565.
- Gamido, H. V. and Gamido, M. V. (2019) ‘Comparative review of the features of automated software testing tools’, *International Journal of Electrical and Computer Engineering*, 9(5), pp. 4473–4478. doi: 10.11591/ijece.v9i5.pp4473-4478.
- Gligoric, M., Eloussi, L. and Marinov, D. (2015) ‘Ekstazi: Lightweight Test Selection’, *Proceedings - International Conference on Software Engineering*, 2, pp. 713–716. doi: 10.1109/ICSE.2015.230.
- Kannan, M; Lokeshwari, K. (2017) ‘Available Online at www.ijarcs.info COMPARISON OF SOFTWARE TESTING TOOLS WITH RESPECT TO TOOLS’, *International Journal of Advanced Research in Computer Science*, 8(0976).
- Legunsen, O. et al. (2016) ‘An extensive study of static regression test selection in modern software evolution’, *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, 13-18-November-2016, pp. 583–594. doi: 10.1145/2950290.2950361.
- Legunsen, O., Shi, A. and Marinov, D. (2017) ‘STARTS: STATic regression test selection’, *32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 3(6), pp. 949–954.

- Rana, K. and Yousuf, M. M. (2019) 'Comparative Analysis of the Automated Penetration Testing Tools', *Journal of Emerging Technologies and Innovative Research*, 5(10), pp. 585–591. Available at: <http://trap.ncirl.ie/4165/>.
- Rosero, R. H., Gómez, O. S. and Rodríguez, G. (2017) 'An approach for regression testing of database applications in incremental development settings', 6th International Conference on Software Process Improvement (CIMPS), pp. 1–4.
- Sandin, E. V., Yassin, N. M. and Mohamad, R. (2016) 'Comparative Evaluation of Automated Unit Testing Tool for PHP', *International Journal of Software Engineering and Technology*, 3(2), pp. 7–11. Available at: <http://ijset.fc.utm.my/ijset/index.php/ijset/article/viewFile/60/31>.
- Satheesh, A. and Singh, M. (2017) 'Comparative Study of Open Source Automated Web Testing Tools: Selenium and Sahi', *Indian Journal of Science and Technology*, 10(13), pp. 1–9. doi: 10.17485/ijst/2017/v10i13/109048.
- Simanca, F. et al. (2019) 'A Solution to Manage the Full Life Cycle of Learning Analytics in a Learning Management System: AnalyTIC', *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, PP(c), p. 1. doi: 10.1109/RITA.2019.2950148.