Software Defect Prediction : Studi Literatur

Hermawan
Magister Informatika Fakultas Ilmu Komputer
Universitas Sriwijaya
Palembang, Sumatera Selatan
hermawanmdp@gmail.com

Santun Irawan
Magister Informatika Fakultas Ilmu Komputer
Universitas Sriwijaya
Palembang, Sumatera Selatan
Santunirawan03@gmail.com

Abstract-Software Defect Prediction, **studi yang berkaitan dengan metode-metode ilmiah untuk memprediksi *error*, *bug*, kegagalan di dalam *software*. Studi ini akan sangat membantu pada tahapan *System Development Life Cycle* (SDLC), terutama pada bagian *testing* perangkat lunak. Metode ini juga bisa digunakan untuk menjamin kualitas pembangunan perangkat lunak, melalui *software metric* bahkan sebelum perangkat lunak tersebut belum dibangun. Sehingga pada akhirnya akan menghemat waktu, usaha dan biaya. Studi selama ini pada umumnya akan menghasilan dataset, metode-metode, dan *framework*, baik yang di implementasikan pada *sample* set atau pada *project* perangkat lunak yang sebenarnya.**

***Keywords* : *Software defect*, *Error*, *Study software defect*.**

I. PENDAHULUAN

Defect bisa dikategorikan sebagai semua *error*, *bug* kesalahan, kecacatan, kegagalan dari sebuah perangkat lunak. Perkembangan pesat dari jumlah perangkat lunak di dunia, dengan banyak tujuan, memberikan sebuah permasalahan baru dalam siklus pengembangan perangkat lunak. Aspek penting dari permasalahan tersebut adalah bagaimana memastikan perangkat lunak tersebut layak untuk dikembangkan, kemudian aspek kualitas tersebut. Kualitas buruk dari sebuah perangkat lunak pada akhirnya akan mengakibatkan *maintenance* perangkat lunak yang lebih rumit, bahkan lebih fatal, mengakibatkan kegagalan secara keseluruhan dari perangkat lunak.

Aspek kualitas tersebut erat kaitannya dengan prediksi *error* dari perangkat lunak. Semakin cepat sebuah *defect* dapat diprediksi, maka semakin cepat pula penanganan dan pencegahan terhadap *defect* tersebut. Estimasi lebih awal dari kemungkinan *error* yang potensial akan membantu dalam proses perencanaan, kontrol, dan eksekusi pada aktivitas pembangunan perangkat lunak [3]. *Output* yang dihasilkan akan sangat membantu pada saat *testing* perangkat lunak pada tahapan SDLC serta memberikan jaminan, baik kepada vendor *software* maupun pada pihak-pihak lain yang berkepentingan.

Studi yang berkembang selama ini dimulai pada awal tahun 90-an dan berkembang pesat akhir-akhir ini. *Trend* studi *software defect* ini biasanya akan menghasilkan data set, metode-metode baru maupun *framework* yang bisa digunakan. Bahkan ada *framework* yang sudah memiliki *Graphical User Interface*. Semua studi tentunya berkaitan dengan studi untuk memastikan kualitas dari perangkat lunak dari kemungkinan *error* yang mungkin terjadi di masa depan. Secara umum, studi tentang *defect* harus difokuskan pada berbagai aspek dari masalah, yang dikategorikan sebagai berikut [2] :

- Pencegahan *Defect*
- Pendeteksian *Defect*
- Koreksi *Defect*

II. TOPIK RISET

Menurut Romi Satria Wahono [1], topik riset berkaitan dengan *software defect,* dibagi dalam 5 kategori yaitu :

1. *Estimation*
   Berkaitan dengan estimasi jumlah *defect* yang ada di perangkat lunak menggunakan algoritma estimasi, dan menggunakan pendekatan statistik.

2. *Associatio*
   Menggunakan algoritma *association rule mining*, yang digunakan untuk menemukan asosiasi *defect.* Biasanya digunakan untuk sebanyak mungkin korelasi antara *defect*. Tentunya, *output* akhirnya bisa digunakan sebagai bahan analisa, untuk mengetahui mengapa beberapa *defect* terjadi berulang-ulang.

3. *Classification*
   Studi yang berkaitan dengan klasifikasi modul di dalam perangkat lunak, menjadi kategori yang cenderung *defect* dan kategori non atau tidak *defect*. Algoritma *classification* yang terkenal menggunakan teknik *Machine Learning*.

4. *Clustering*
   Studi yang menggunakan algoritma *clustering* dari komunitas data mining, untuk menentukan pengelompokan (*clustering*) *area defect* yang ada di perangkat lunak.

5. Analisis Dataset
   Analisis terhadap dataset *defect* yang sudah ada. Bisa dilakukan dengan berbagai pendekatan algoritma dan dari berbagai sudut pandang.

Berdasarkan hasil *survey* dari Romi Satria Wahono [1], topik yang popular dikerjakan oleh para peneliti adalah kategori *Classification.*

III. *SOFTWARE DEFECT PREDICTION MODEL*

Trend penelitian yang berkaitan dengan *software*, biasa dikategorikan ke dalam beberapa kelompok sebagai berikut :

3.1 Model Prediksi Menggunakan *Size* dan Kompleksitas *Metrics*

Model prediksi ini bisa juga dikategorikan sebagai metode tradisional. Model ini akan menggunakan kode program sebagai basis untuk prediksi *defect*. Konsep yang umum digunakan adalah *Lines of Code* (LOC), algoritma kompleksitas McCabes's Cyclomatic, dan *function point* yang dikembangkan oleh Alberecht di tahun 1979 untuk menggantikan LOC. Model ini mengasumsikan bahwa *defect* terjadi dikarenakan ukuran dan kompleksitas dari sebuah perangkat lunak. Model ini mengabaikan faktor-faktor lain di luar ukuran dan kompleksitas, seperti faktor kesalahan manusia, kemudian *design* awal yang buruk.

3.2 *Machine Learning Based Model*

Dengan menggunakan metode ini, penelitian bisa diperluas untuk menjawab berbagai aspek penyebab dari semua kegagalan, *error* yang terjadi di dalam perangkat lunak. Model ini biasanya menggunakan teknik *soft computing* untuk memprediksi *defec*t.Beberapa algoritma yang digunakan di adalah *artificial neural network, concept learning*, *Bayesian Belief networks, reinforcement learning, genetic algorithms, genetic programming, instance-based learning, decision tree, inductive based programming, dan analytical learning* [2]. Beberapa algoritma lain yang populer digunakan adalah *ensemble classifier, random forest, naïve Bayes Classifier, support vector machine, multi layer perceptron, radial basis function networks.*

3.3 *Defect Density Prediction Model*

Defect Density adalah pengukuran dari semua *defec*t dibagi dengan ukuran entitas perangkat lunak. Beberapa model yang popular adalah *Constructive Quality Modelling for Defect Densitiy Prediction* (COQUALMO) dan *Defect Prediction Model Based on Six Sigma Metrics.*

IV. *FRAMEWORK SOFTWARE DEFECT PREDICTION*

Penelitian terhadap *software defect* yang berkembang pesat tentunya membawa kontribusi besar bagi siklus SDLC. Perkembangan selanjutnya adalah pembuatan *framework*, yang memberikan efektivitas dan kemudahan dalam proses prediksi *defect* itu sendiri. Bahkan sudah pernah ada *framework* yang menyertakan *Graphical User Interface* (GUI) dengan menggunakan jaringan saraf tiruan. Beberapa *framework* populer yang dikembangkan adalah [1] :

1. Menzies et al's Framework
   Menzies mengajukan *framework* studi untuk mengkomparasi dua teknik algoritma *classification* untuk memprediksi komponen *software* yang mengandung *defect* pada dataset NASA MDP *repository.*

2. Lessmann et al's Framework
   Framework lessman merupakan perkembangan dari *framework* Menzies. Akan tetapi Lessman tidak melakukan seleksi *attribut* ketika membuat model prediksi. Lessman mengajukan sebuah *framework* dengan membuat *benchmark* pada algoritma *classification* pada berbagai dataset dan mengamati bahwa metode *classification* menyediakan akurasi yang baik pada hasil prediksi dan mendukung untuk *metric based application* [4].

3. Song et' al Framework
   Masih berdasarkan perkembangan dari Menzies *Framework*, *Song* membangun *framework* untuk *software defect* untuk tujuan umum, yang fokus pada dua bagian : skema evaluasi dan prediksi *defect*. Skema evaluasi digunakan untuk mengevaluasi kinerja skema learning. Sementara prediksi *defect*, *focus* dalam membangun *final predictor* menggunakan data historis pada skema *learning*. Baru setelah itu *final predictor* yang akan mendeteksi bagian dari perangkat lunak yang rawan terhadap *defect* [1].

V. DATASET

Semua penelitian terhadap *defect* perangkat lunak, pastinya akan membutuhkan dataset sebagai sumber data. Secara umum, bisa dibedakan menjadi dua kategori utama. Pertama, dataset *public* yang bisa diakses umum secara *online*. Kedua, dataset yang sifatnya *private*, yang tidak dipublikasikan, dan biasanya milik perusahaan tertentu. Sebaiknya terhadap penelitian menggunakan dataset *public*, dikarenakan data tersebut yang banyak digunakan oleh peneliti. Apabila menggunakan dataset *private* sebaiknya bisa dipublikasikan kepada umum, sehingga bisa dikomparasi dan digunakan oleh peneliti lain. Hampir penelitian *defect* menggunakan dataset yang berasal dari PROMISE repository, tepatnya NASA *Metrics Data Program* (NASA MDP).

VI. ISU *SOFTWARE DEFECT*

Selain manfaat yang diberikan oleh *software defect prediction*, ada beberapa isu yang tentunya berpengaruh terhadap performance.Beberapa isi berkaitan dengan *software defect prediction* [5] :

1. Relasi antara attribut dan *fault /defect*.
   Para peneliti tidak mampu untuk mengidentifkasi subset dari attribute yang menjadi faktur substansial bagi sebuah modul yang bisa dikategorikan sebagai fault/ defect. Belum ada yang bisa menunjukan hubungan antara attribute dan fault.

2. Tidak ada standar pengukuran untuk *performance*
   Dikarenakan tidak standar *performace* yang ada di dunia, maka banyak sekali laporan performace yang tidak

konsisten di berbagai area subjek penelitan *software defect prediction*.

3. Isu terhadap *Cross Project Defect Prediction*
   Beberapa peneliti, masih menggunakan data local /*private* yang didapatkan dari beberapa *project.* Data ini biasa digunakan untuk penelitian lain yang punya *project* yang hampir sama. Data ini bisa jadi sudah tidak ada lagi. Para peneliti selanjutnya, kemudian menemukan solusi untuk membuat *cross project defect prediction*, dimana model *cross project defect prediction model* dibangun pada suatu *project* dan di observasi bersama. Dimana model tersebut dibangun pada satu *project* dan digunakan pada *project* lain, dan bisa diobservasi bersama-sama. Akan tetapi *performace* dari model yang dibangun di atas berbagai data perusahaan yang berbeda-beda tentunya tidak menjanjikan sama sekali.

4. Tidak ada standar *framework* umum.
   Banyak sekali peneliti yang menggunakan banyak metode, banyak teknik dan juga banyak dataset. Akan tetapa samapai sekarang tidak ada standar umum / *framework* yang bisa dipakai sebagai *benchmark* standar dari penelitian untuk dipakai di berbagai perusahaan dan tentunya perangkat lunak. Walaupun memang sudah ada *framework,* akan tetapi belum diakui secara internasional sebagai general *framework*.

5. Nilai ekonomi dari *software defect prediction*
   Banyak penelitan belum memperhatikan aspek ekonomis dari *software defect prediction*.

6. *Problem class imbalance*
   Efisiensi dari *software defect prediction*, dibentuk dari kelompok distribusi dari *training* data. Kelompok distribusi tersebut dibentuk dari jumlah *instance* dari setiap kelompok di dalam *training* data. Apabila jumlah *instance* yang dimiliki dari satu kelompok lebih besar dari kelompok lain. Maka disimpulkan bahwa ada masalah ketidakseimbangan kelompok (*class imbalance*). Efeknya akan mempengaruh hasil dari *software defect prediction* secara keseluruhan.

VII. KESIMPULAN

Software Engineering, selalu dikaitan dengan proses dan tahapan dalam pembangunan perangkat. Merupakan suatu keharusan, para *developer* perangkat lunak bisa memastikan kualitas perangkat lunak yang di akan atau sedang dikembangkan. Di samping, mempertimbangkan waktu, usaha, dan biaya untuk mengembangkan perangkat lunak. *Software Defect Prediction* adalah metode untuk memprediksi *bug*, *error*, cacat yang mungkin ada di perangkat lunak tersebut. Informasi tersebut bisa menjadi pertimbangan apakah perangkat lunak tersebut layak untuk dikembangkan, atau menjadi informasi awal untuk mengetahui kelemahan dari perangkat lunak. Tidak boleh dilupakan juga, tahapan SDLC adalah tahapan yang terjadi pada dunia industri perangkat lunak. Sehingga semua penelitan yang ada juga harus bisa diaplikasikan, atau setidaknya membantu di dalam dunia industri perangkat lunak. Akan tetapi, semua studi pada *software defect prediction* pada dunia industri *software* di dunia nyata masih bisa dikatakan sangat jarang [6].

REFERENSI

[1] Romi Satria Wahono, "A Systematic Literature Review of Software Defect Prediction :Research Trends, Datasets, Methods, and Framework", Journal of Software Engineering, Vol. 1, No. 1, April 2015

[2] Mrinal Singh Rawat and Sanjay Kumar Dubey, *Software Defect Prediction Models For Quality Improvement: A Literature Study* , IJCSI International Journal Of Computer Sciences Issues, Vol 9, Issue 5, No.2, September 2012.

[3] Sarita Signgh Chauhan, and Usaha Husain, "An AI Approach for Software Defect Prediction – A Review of Literature", International Journal Of Research and Development in Apllied Scienc and Enginerring (IJRDASE) Volume 1, Issue 2, August 2015

[4] Saiqa Aleem and Luiz Fernando Capretz and Faheem Ahmed "Benchmarking Machine Learning Techniques for Software Defect Detection", International Journal Of Software Engineering & Application (IJSEA) Vol 6, No. 3, May 2015

[5] Ishani Arora and VIvek Tetarwal, dan Anju Saham, "Open Issues in Software Defects", Procedia Computer Science 46 (2015) 906 – 912.

[6] Jaroslaw HRYSZKO dan Lech MADEYSKI, "Bottleneck in Software Defect Prediction Implementation in Industrial Procets", foundations of computing and decision sciences Vol. 40 No. 1 (2015).

[7] Vipul Vashisht and Manohar Lal and GS. Sureshchandar, Suraj Kamya, "A Framework for Software Defect Prediction Using Neural Network", Journal of Software Engineering and Application, 2015, 8, 384-394

ISBN: 979-587-573-6