BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 2, Agustus Tahun 2024
ISSN 3024-8248 (media online)
Hal 300-305

Penggunaan *Machine Learning* untuk Memprediksi *Defect* pada Pengembangan Perangkat Lunak

Gilang Saputro[1*], Kinta Aurora[2], Toto Winarko[3], Andara Shayla[4], Aries Saifudin[5]

[1-5]Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia
Email: [1*]gilangsaputro204@gmail.com, [2]kintaaurora12@gmail.com, [3]totowianrko70@gmail.com, [4]andarashayla121 @gmail.com, [5]aries.saifudin@unpam.ac.id
(* : coressponding author)

Abstrak− Penelitian ini mengaplikasikan machine learning untuk memprediksi defect dalam pengembangan perangkat lunak, menggunakan model Random Forest, SVM, dan Neural Networks. Hasil menunjukkan akurasi tinggi dalam identifikasi cacat, memungkinkan perbaikan lebih awal dan peningkatan efisiensi serta kualitas produk. Tantangan meliputi kebutuhan dataset representatif dan penyesuaian parameter model. Secara keseluruhan, metode ini menawarkan solusi inovatif untuk peningkatan kualitas pengembangan perangkat lunak.

Kata Kunci: *Machine Learning,* Prediksi *Defect,* Pengembangan Perangkat Lunak, Random Forest, *Support Vector Machine, Neural Networks,* Kualitas Perangkat Lunak, Efisiensi Pengembangan.

Abstract− *This research applies machine learning to predict defects in software development, using Random Forest, SVM, and Neural Networks models. Results show high accuracy in defect identification, enabling earlier repairs and increased efficiency and product quality. Challenges include the need for representative datasets and fine-tuning of model parameters. Overall, these methods offer innovative solutions to improve the quality of software development.*

Keywords*: Machine Learning, Defect Prediction, Software Development, Random Forest, Support Vector Machine, Neural Networks, Software Quality, Development Efficiency.*

1. PENDAHULUAN

Pentingnya Obyek/Aktivitas yang Akan Dikomputerisasi

Dalam industri pengembangan perangkat lunak, prediksi dan identifikasi defect atau cacat pada tahap awal merupakan aspek krusial yang dapat menentukan keberhasilan proyek. Defect yang terdeteksi pada tahap akhir pengembangan atau setelah produk diluncurkan dapat menyebabkan peningkatan biaya perbaikan, penurunan kualitas produk, dan kehilangan kepercayaan pengguna. Oleh karena itu, mengimplementasikan teknologi yang mampu memprediksi defect pada perangkat lunak sejak awal merupakan kebutuhan mendesak untuk meningkatkan efisiensi dan efektivitas proses pengembangan.

Obyek/Aktivitas Saat Ini yang Berjalan

Saat ini, identifikasi defect pada pengembangan perangkat lunak sering kali dilakukan melalui proses manual, seperti code review dan pengujian yang intensif. Teknik ini, meskipun efektif, memerlukan waktu dan sumber daya yang signifikan. Selain itu, banyak organisasi yang menggunakan alat otomatisasi pengujian, tetapi alat ini lebih sering digunakan untuk mendeteksi bug yang sudah ada daripada memprediksi kemungkinan munculnya defect di masa depan.

Masalah yang Akan Diselesaikan dengan Mengembangkan Software

Masalah utama yang dihadapi dalam pengembangan perangkat lunak adalah keterlambatan dalam mendeteksi defect, yang menyebabkan peningkatan biaya dan penurunan kualitas produk akhir. Dengan memprediksi defect lebih awal, tim pengembang dapat mengalokasikan sumber daya dengan lebih efektif, memperbaiki kode yang bermasalah sebelum menjadi lebih kompleks, dan secara keseluruhan meningkatkan proses pengembangan. Menggunakan machine learning untuk memprediksi defect dapat

Buletin Ilmiah
Ilmu Komputer dan Multimedia
BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 2, Agustus Tahun 2024
ISSN 3024-8248 (media online)
Hal 300-305

mengatasi masalah ini dengan memberikan wawasan yang didasarkan pada analisis data historis dan pola-pola yang mungkin tidak terlihat oleh manusia.

Metode/Model yang Akan Digunakan untuk Mengembangkan Software dan Alasannya

Untuk mengembangkan software prediksi defect ini, akan digunakan metode machine learning, khususnya model supervised learning seperti Random Forest, Support Vector Machine (SVM), dan Neural Networks. Metode supervised learning dipilih karena ketersediaan data historis yang dapat digunakan untuk melatih model dalam memprediksi defect berdasarkan fitur-fitur tertentu seperti kompleksitas kode, riwayat perubahan, dan hasil pengujian sebelumnya. Random Forest dipilih karena kemampuannya dalam menangani data yang tidak seimbang dan mengurangi overfitting. SVM dipilih karena performanya yang baik dalam high-dimensional spaces, dan Neural Networks karena fleksibilitasnya dalam menangkap pola-pola kompleks dalam data.

Rencana Pelaksanaan Pengembangan Software

Pengembangan software prediksi defect akan dilaksanakan dalam beberapa tahap sebagai berikut:

a. Pengumpulan Data: Mengumpulkan data historis pengembangan perangkat lunak, termasuk metrik kode, riwayat perubahan, dan hasil pengujian.
b. Preprocessing Data: Membersihkan dan menyiapkan data untuk pelatihan model machine learning, termasuk penanganan data yang hilang dan normalisasi fitur.
c. Pemilihan dan Pelatihan Model: Menggunakan beberapa algoritma machine learning untuk melatih model pada data yang telah dipreproses dan melakukan evaluasi performa model menggunakan teknik cross-validation.
d. Validasi dan Pengujian Model: Menguji model pada data uji yang terpisah untuk memastikan akurasi dan generalisasi model.
e. Implementasi dan Integrasi: Mengintegrasikan model prediksi defect ke dalam pipeline pengembangan perangkat lunak yang ada.
f. Pemantauan dan Pemeliharaan: Melakukan pemantauan performa model secara berkala dan melakukan penyesuaian jika diperlukan memastikan model tetap efektif seiring waktu.

Dengan rencana ini, diharapkan software prediksi defect berbasis machine learning dapat memberikan kontribusi signifikan dalam meningkatkan kualitas dan efisiensi pengembangan perangkat lunak.

2. METODE PENELITIAN

2.1 Metodologi Pengembangan Software

2.1.1 Teori Obyek/Aktivitas yang Akan Dikomputerisasi

Pada pengembangan perangkat lunak untuk memprediksi defect menggunakan machine learning, obyek utama yang akan dikomputerisasi adalah proses identifikasi dan prediksi cacat atau defect dalam kode perangkat lunak. Secara teori, proses ini melibatkan penggunaan data historis tentang pengembangan perangkat lunak, termasuk metrik kode seperti kompleksitas, ukuran, dan riwayat perubahan, serta hasil pengujian untuk melatih model machine learning. Tujuannya adalah untuk mengembangkan model yang dapat memprediksi dengan akurat kemungkinan munculnya defect berdasarkan fitur-fitur ini.

2.1.2 Metode/Model yang Digunakan Berdasarkan Teori

Metode yang akan digunakan adalah supervised learning, yang memanfaatkan dataset yang telah diberi label (yaitu, data yang sudah diketahui outputnya) untuk melatih model. Model-machine learning yang dipilih termasuk:

Buletin Ilmiah
Ilmu Komputer dan Multimedia
BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 2, Agustus Tahun 2024
ISSN 3024-8248 (media online)
Hal 300-305

a. Random Forest: Model ini dipilih karena kemampuannya dalam menangani data yang tidak seimbang dan mengurangi overfitting, yang sering kali terjadi dalam dataset pengembangan perangkat lunak.
b. Support Vector Machine (SVM): SVM dipilih karena performanya yang baik dalam menangani high-dimensional spaces, yang cocok untuk fitur-fitur kompleks seperti yang ditemukan dalam kode perangkat lunak.
c. Neural Networks: Neural Networks dipilih karena kemampuannya untuk mempelajari pola-pola kompleks dalam data, yang dapat membantu dalam memprediksi defect berdasarkan pola-pola yang rumit dan tidak terstruktur dalam kode.

2.1.3 Rencana dan Rancangan Pengembangan Software Berdasarkan Teori

Pengembangan software prediksi defect akan mengikuti pendekatan berikut:

1. Pengumpulan Data: Mengumpulkan dataset historis yang mencakup metrik kode, riwayat perubahan, dan hasil pengujian dari proyek-proyek pengembangan perangkat lunak sebelumnya.
2. Preprocessing Data: Membersihkan dan menyiapkan data untuk pelatihan model machine learning, termasuk penanganan data yang hilang, normalisasi fitur, dan pemrosesan lanjutan seperti reduksi dimensi jika diperlukan.
3. Pemilihan dan Pelatihan Model: Memilih beberapa model-machine learning seperti Random Forest, SVM, dan Neural Networks untuk dilatih pada data yang telah dipreproses. Evaluasi performa model dilakukan menggunakan teknik cross-validation untuk memastikan kehandalan dan akurasi.
4. Validasi dan Pengujian Model: Menguji model pada dataset uji yang terpisah untuk memvalidasi kemampuan prediksi dan generalisasi model terhadap data baru.
5. mplementasi dan Integrasi: Mengintegrasikan model-machine learning yang telah terlatih ke dalam pipeline atau alur kerja pengembangan perangkat lunak yang ada, sehingga dapat digunakan secara rutin dalam proses pengembangan.
6. Pemantauan dan Pemeliharaan: Melakukan pemantauan secara berkala terhadap performa model dan melakukan penyesuaian jika diperlukan untuk memastikan model tetap efektif seiring waktu dan perubahan dalam lingkungan pengembangan perangkat lunak.

Dengan mengikuti rencana dan rancangan ini berdasarkan teori yang telah dijelaskan, diharapkan pengembangan software prediksi defect berbasis machine learning dapat meningkatkan kualitas dan efisiensi dalam pengembangan perangkat lunak.

3. ANALISA DAN PEMBAHASAN

Pada tahap ini, hasil dari penerapan metode dan model yang telah dirancang dalam bab 2 (Metodologi) akan dibahas secara detail berdasarkan evaluasi dan pengujian yang dilakukan.

a. Pengumpulan Data: Dataset historis yang mencakup metrik kode, riwayat perubahan, dan hasil pengujian dari proyek-proyek pengembangan perangkat lunak telah berhasil dikumpulkan dan disiapkan untuk analisis lebih lanjut.
b. Preprocessing Data: Data telah dibersihkan dari nilai-nilai yang hilang dan fitur-fitur telah dinormalisasi untuk mempersiapkan mereka untuk pelatihan model machine learning. Langkah-langkah preprocessing ini sangat penting untuk memastikan kualitas dan akurasi model yang akan dikembangkan.
c. Pemilihan dan Pelatihan Model: Beberapa model-machine learning, termasuk Random Forest, SVM, dan Neural Networks, telah dipilih dan dilatih pada data yang telah dipreproses. Proses pelatihan ini mencakup penggunaan teknik cross-validation untuk mengukur dan memvalidasi kinerja model.
d. Validasi dan Pengujian Model: Model yang telah dilatih dievaluasi menggunakan dataset uji yang terpisah untuk memvalidasi kemampuan mereka dalam memprediksi defect.

Buletin Ilmiah
Ilmu Komputer dan Multimedia
BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 2, Agustus Tahun 2024
ISSN 3024-8248 (media online)
Hal 300-305

Evaluasi ini mencakup pengukuran akurasi, presisi, recall, dan metrik evaluasi lainnya untuk memastikan kehandalan dan generalisasi model terhadap data baru.

3.1 Pembahasan/Analisis

1. Performa Model: Hasil evaluasi menunjukkan bahwa semua model-machine learning yang dipilih (Random Forest, SVM, dan Neural Networks) memiliki tingkat akurasi yang tinggi dalam memprediksi kemungkinan munculnya defect dalam kode perangkat lunak. Random Forest menunjukkan kinerja yang baik dalam menangani dataset yang tidak seimbang dan mencegah overfitting, sementara SVM memberikan presisi yang baik dalam menangani fitur-fitur kompleks. Neural Networks, meskipun kompleks, mampu mengenali pola-pola yang rumit dalam data.
2. Kelebihan dan Keterbatasan: Penggunaan metode-machine learning ini membawa keuntungan signifikan dalam meningkatkan efisiensi proses identifikasi defect dibandingkan dengan metode manual tradisional. Namun, keterbatasan yang mungkin timbul termasuk kebutuhan akan data yang cukup dan representatif untuk pelatihan model, serta kompleksitas dalam mengatur parameter dan tuning model-machine learning.
3. Implikasi Praktis: Implementasi model-machine learning ini dapat secara signifikan mengurangi biaya perbaikan dan peningkatan kualitas produk perangkat lunak dengan memungkinkan tim pengembang untuk mendeteksi dan memperbaiki defect lebih awal dalam siklus pengembangan.
4. Kesimpulan: Berdasarkan hasil dan pembahasan ini, dapat disimpulkan bahwa pengembangan software prediksi defect berbasis machine learning sesuai dengan desain yang telah dirancang mampu memberikan kontribusi positif dalam meningkatkan kualitas dan efisiensi pengembangan perangkat lunak.

Dengan demikian, penelitian ini tidak hanya mengonfirmasi keefektifan model-machine learning dalam prediksi defect, tetapi juga memberikan landasan pengembangan lebih lanjut dalam bidang ini.

4. KESIMPULAN

4.1 Kesimpulan

Berdasarkan analisis hasil penerapan metode untuk menyelesaikan masalah prediksi defect dalam pengembangan perangkat lunak menggunakan machine learning, dapat diambil kesimpulan sebagai berikut:

1. Efektivitas Metode: Model-machine learning seperti Random Forest, SVM, dan Neural Networks telah terbukti efektif dalam memprediksi kemungkinan munculnya defect dalam kode perangkat lunak. Hasil evaluasi menunjukkan tingkat akurasi yang tinggi dalam mengidentifikasi potensi cacat, yang dapat membantu tim pengembang untuk mengambil tindakan korektif lebih awal.
2. Keuntungan Signifikan: Implementasi metode-machine learning ini memberikan keuntungan signifikan dibandingkan dengan metode manual tradisional. Kemampuan untuk mendeteksi dan memperbaiki defect lebih awal dalam siklus pengembangan tidak hanya mengurangi biaya perbaikan, tetapi juga meningkatkan kualitas produk akhir.
3. Keterbatasan dan Tantangan: Meskipun demikian, penggunaan model-machine learning juga menghadapi tantangan, seperti kebutuhan akan dataset yang cukup dan representatif untuk pelatihan model, serta kompleksitas dalam penyesuaian parameter dan tuning model untuk mencapai performa optimal.
4. Implikasi Praktis: Secara praktis, penerapan teknologi ini dapat meningkatkan efisiensi proses pengembangan perangkat lunak secara keseluruhan. Dengan menggunakan model-machine learning untuk memprediksi defect, organisasi dapat menghemat waktu dan sumber daya yang berharga, sambil meningkatkan kepuasan pengguna akhir dengan produk yang lebih andal dan berkualitas.

Buletin Ilmiah
Ilmu Komputer dan Multimedia
BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 2, Agustus Tahun 2024
ISSN 3024-8248 (media online)
Hal 300-305

Dengan demikian, pengembangan software prediksi defect berbasis machine learning bukan hanya memberikan solusi yang inovatif dalam mengatasi tantangan dalam pengembangan perangkat lunak, tetapi juga membuka jalan bagi pengembangan teknologi yang lebih lanjut dalam mendukung industri teknologi informasi dan komputerisasi di masa depan.

4.2 Saran

Untuk penelitian selanjutnya, disarankan untuk mempertimbangkan beberapa hal berikut agar mendapatkan hasil yang lebih baik:

1. Pengumpulan Data yang Lebih Diversifikasi: Mengumpulkan dataset yang lebih diversifikasi dari berbagai proyek pengembangan perangkat lunak, termasuk berbagai domain dan ukuran proyek yang berbeda, untuk meningkatkan representasi dan generalisasi model.
2. Penelitian Mengenai Feature Engineering: Melakukan penelitian lebih lanjut mengenai feature engineering yang lebih maju dan relevan dalam konteks pengembangan perangkat lunak, untuk meningkatkan kualitas input data yang digunakan dalam pelatihan model.
3. Eksplorasi Model-Machine Learning Lainnya: Menggunakan dan membandingkan dengan model-machine learning lainnya yang baru dan inovatif dalam memprediksi defect, untuk mengeksplorasi kemungkinan pendekatan yang lebih efisien dan akurat.
4. Implementasi Metode Ensemble: Menerapkan metode ensemble dari beberapa model-machine learning untuk meningkatkan kehandalan prediksi, dengan menggabungkan kelebihan masing-masing model dalam memproses dan menganalisis data.
5. Analisis Efek Keamanan dan Performa: Melakukan analisis lebih lanjut terhadap efek keamanan dan performa dari model-machine learning yang diimplementasikan dalam lingkungan pengembangan perangkat lunak yang sebenarnya.
6. Studi Kasus Implementasi di Industri: Melakukan studi kasus implementasi nyata di industri, untuk mengukur dampak langsung dari penggunaan teknologi prediksi defect berbasis machine learning dalam pengembangan perangkat lunak.

Dengan mempertimbangkan saran-saran ini, diharapkan penelitian selanjutnya dapat menghasilkan pembaruan dan kontribusi yang signifikan dalam pengembangan teknologi prediksi defect untuk industri teknologi informasi dan komputerisasi.

REFERENCES

Bahtiar, Agus, Mulyawan, Suryani, and Dindin Firmansyah. “Prediksi Cacat Sofware Menggunakan Algoritma C4.5 Berbasis Particle Swarm Optimization” XX, no. Xx (2019): 1–6. https://jurnal.kopertipindonesia.or.id/index.php/kopertip/article/view/74/45.

Erna, Irawan, and Wahono Romi Satria. “Penggunaan Random Under Sampling Untuk Penanganan Ketidakseimbangan Kelas Pada Prediksi Cacat Software Berbasis Neural Network.” *Journal of Software Engineering* 1, no. 2 (2015): 92–100.

Fahdia, M R, and R E Indrajit. “Linear Regression Dengan Pembobotan Atribut Dengan Metode Pso Untuk Software Defect Prediction.” *Prosiding Semnastek*, no. November (2017): 1–2. https://jurnal.umj.ac.id/index.php/semnastek/article/view/2020%0Ahttps://jurnal.umj.ac.id/index.php/semnastek/article/download/2020/1662.

GOOD, GOOLMAN. “**済無**No Title No Title No Title.” *Angewandte Chemie International Edition, 6(11), 951–952.* 1, no. April (2015).

Hardoni, Andre, Dian Palupi Rini, and Sukemi Sukemi. “Integrasi SMOTE Pada Naive Bayes Dan Logistic Regression Berbasis Particle Swarm Optimization Untuk Prediksi Cacat Perangkat Lunak.” *Jurnal Media Informatika Budidarma* 5, no. 1 (2021): 233. https://doi.org/10.30865/mib.v5i1.2616.

Prasetyo, Rizal, Imam Nawawi, Ahmad Fauzi, and Ginabila Ginabila. “Komparasi Algoritma Logistic

BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia
Volume 2, No. 2, Agustus Tahun 2024
ISSN 3024-8248 (media online)
Hal 300-305

Regression Dan Random Forest Pada Prediksi Cacat Software." *Jurnal Teknik Informatika UNIKA Santo Thomas* 06, no. Siringoringo 2017 (2021): 275–81. https://doi.org/10.54367/jtiust.v6i2.1522.

Putri, Sukmawati Anggraeni, and Dewi Larasati. "Penerapan Feature Selection Pada Bayesian Network Untuk." *PILAR Nusa Mandiri* 13, no. 2 (2017): 275–80.

Suardana, Putu Agus. "Penerapan Metode Rayleigh Dalam Prediksi Keandalan Pada Aplikasi Berbasis Web" 2, no. 2 (2015): 5928–35.

Wicaksono, Aji. "Prediksi Dan Deteksi Bug Pada Software Menggunakan Pendekatan Machine Learning: Machine Learning." *JURNAL SIGN IN: Jurnal Ilmiah Sistem Informasi Dan Informatika* 2, no. 2 (2023): 14–17.

Wicaksono, Satrio Agung, Daniel Oranova S., and Sarwosri Sarwosri. "Pembangunan Model Prediksi Defect Menggunakan Metode Ensemble Decision Tree Dan Cost Sensitive Learning." *Jurnal EECCIS* 4, no. 1 (2010): 1–7.