

BAB V PENUTUP

5.1 Kesimpulan

Penelitian ini bertujuan menguji kemampuan algoritma *Random Forest* dalam memprediksi kecacatan perangkat lunak menggunakan pendekatan *Just-in-Time* pada dataset ApacheJIT, dengan memperhatikan aspek waktu dalam penilaian model. Kesimpulan yang diperoleh berdasarkan hasil penelitian ini dirancang untuk menjawab masalah yang telah ditetapkan di Bab I.

Rumusan masalah pertama berhubungan dengan kemampuan algoritma *Random Forest* dalam memprediksi *commit* yang mengandung kecacatan. Hasil penelitian menunjukkan bahwa algoritma ini mampu memberikan hasil prediksi yang kompetitif di tingkat *commit*, terutama pada metrik *Recall*. Hasil ini menunjukkan bahwa model dapat dengan baik mengidentifikasi *commit* yang berpotensi mengandung kecacatan, sesuai dengan karakteristik *Just-in-Time Defect Prediction* yang menekankan pentingnya mendeteksi kecacatan sejak dini.

Rumusan masalah kedua berhubungan dengan dampak cara membagi data terhadap hasil evaluasi model. Hasil eksperimen menunjukkan bahwa skema *Random Split* memberikan hasil kinerja yang lebih baik dibandingkan skema *Chronological Split*. Namun, evaluasi menggunakan skema *Chronological Split* memberikan estimasi kinerja yang lebih realistis, karena mempertimbangkan urutan waktu *commit* dan mencegah kebocoran informasi waktu. Temuan ini menunjukkan bahwa penggunaan skema *Chronological Split* lebih tepat untuk mengevaluasi kinerja model dalam kondisi pengembangan perangkat lunak nyata.

Rumusan masalah ketiga berkaitan dengan faktor-faktor yang mempengaruhi hasil prediksi kecacatan. Analisis *Feature Importance* menunjukkan bahwa metrik perubahan kode, terutama jumlah baris kode yang ditambahkan, memiliki kontribusi terbesar dalam proses prediksi. Temuan ini menunjukkan bahwa perubahan kode dalam skala besar berkaitan erat dengan peningkatan risiko terjadinya kecacatan perangkat lunak pada tingkat *commit*.

Berdasarkan keseluruhan hasil penelitian, kontribusi utama penelitian ini adalah penyajian evaluasi prediksi kecacatan *Just-in-Time* yang lebih realistis dengan menggunakan skema *Chronological Split* sebagai pendekatan utama dalam evaluasi, serta penyediaan analisis interpretatif melalui pentingnya fitur untuk memahami faktor-faktor utama penyebab kecacatan. Kontribusi ini memperkuat pemahaman tentang pentingnya aspek waktu dan kemampuan interpretasi model dalam penerapan *Just-in-Time Defect Prediction*.

5.2 Saran

Saran yang diusulkan dalam penelitian ini terbagi menjadi dua jenis, yaitu saran praktis dan saran akademik. Saran praktis ditujukan kepada para praktisi dan pengembang perangkat lunak. Implementasi sistem prediksi cacat perangkat lunak dapat dikembangkan dalam bentuk aplikasi pendukung dalam proses *quality assurance software*. Dengan demikian, model prediksi dapat terintegrasi langsung ke dalam tahap *code review* atau *continuous integration*. Penggunaan sistem ini diharapkan mampu membantu para pengembang dalam mengenali *commit* yang memiliki risiko tinggi mengandung cacat lebih awal.

Saran akademik ditujukan untuk penelitian selanjutnya. Penelitian lanjutan dapat menguji model prediksi cacat perangkat lunak dengan menggunakan algoritma *ensemble* lain seperti *XGBoost* atau *LightGBM*, sehingga membandingkan kinerja dan tingkat interpretabilitas model dengan *Random Forest*. Penggunaan dataset lain yang memiliki ukuran dan karakteristik berbeda, seperti dataset *Mozilla* atau *Eclipse*, juga dapat dilakukan untuk menguji kemampuan model dalam berbagai domain perangkat lunak. Selain itu, validasi model dapat diperluas dengan menggunakan pendekatan *time-based cross-validation* untuk menangani perubahan distribusi data seiring waktu (*concept drift*) secara lebih menyeluruh. Penelitian lanjutan juga dapat difokuskan pada peningkatan interpretabilitas model melalui visualisasi *Feature Importance* atau metode *explainability* yang tidak bergantung pada model, sehingga hasil prediksi dapat lebih mudah dipahami oleh para pengembang perangkat lunak.