INTISARI

SEMBADA MAJU, Sistem Basis Data Masyarakat Jumeneng, adalah aplikasi kependudukan berbasis desktop yang dibuat untuk menunjang aktivitas administrasi kependudukan Dukuh Jumeneng. Aplikasi ini tidak tersambung internet mengingat risiko terjadinya pencurian data. Penelitian ini dilakukan untuk mengurangi risiko tersebut dengan mendeteksi SQL *injection* menggunakan NLP sebelum eksekusi. Pada penelitian ini, empat jenis model klasifikasi dan tiga macam *pre-processing* akan dievaluasi dalam mendeteksi SQL *injection*. Model klasifikasi yang dimaksud adalah *logistic regression*, BiLSTM, TextCNN, dan ResNet. Sedangkan *pre-processing* yang dimaksud adalah generalisasi, eliminasi, dan tanpa *pre-processing*. Tujuannya untuk melihat performa tiap model klasifikasi dalam mendeteksi SQL *injection* dan pengaruh *pre-processing* berbeda terhadap performa dan waktu pemrosesan dari setiap model klasifikasi.

Setiap kombinasi model klasifikasi dengan metode *pre-processing* akan dilatih dengan *dataset* kecil berisi SQL *injection* dan *statement* biasa. Performa dan waktu pelatihan setiap kombinasi akan dievaluasi pada tahap ini. Model yang sudah terlatih kemudian akan divalidasi dengan melakukan klasifikasi pada *dataset* yang lebih besar dan sepenuhnya terdiri atas SQL *injection.* Validasi dilakukan untuk mengetahui apakah model bisa mengidentifikasi SQL *injection* yang tidak terdapat pada *dataset* latihan atau tidak. Akurasi dan rata-rata waktu pemrosesan dari *pre-processing* hingga didapatkan klasifikasi oleh setiap kombinasi akan dievaluasi pada tahap ini.

Hasil evaluasi menunjukkan metode *pre-processing* generalisasi cenderung membantu model dalam mencapai performa terbaik. Pada tahap pelatihan, hampir setiap model mencapai akurasi ~99%, kecuali ResNet dengan metode *pre-processing* eliminasi yang mengalami kerusakan dan mengklasifikasi semua masukkan sebagai SQL *injection* sehingga akurasinya hanya mencapai ~34%. Performa setiap model pada tahap validasi mengalami penurunan. *Logistic regression* mengalami penurunan akurasi menjadi ~19%, ini menunjukkan model ini tidak dapat mengolah masukkan di luar *dataset* pada pelatihan. Di sisi lain, model *deep learning* menunjukkan performa yang lebih baik. BiLSTM menunjukkan akurasi pada kisaran 83-91%, sedangkan ResNet menunjukkan akurasi pada kisaran 78-92%. Pada kedua model ini, performa terbaik dicapai dengan melakukan *pre-processing* generalisasi. Pada model TextCNN, performa antar metode *pre-processing* tidak mengalami perubahan yang serupa, akurasi yang ditunjukkan berada pada kisaran 84-86%.

Dapat disimpulkan bahwa model *deep learning* seperti BiLSTM, TextCNN, dan ResNet dapat mengolah masukkan, bahkan yang tidak ada pada saat pelatihan, tidak seperti model *machine learning* tradisional seperti *logistic regression*. Dibandingkan *pre-processing* lainnya, generalisasi secara konsisten meningkatkan akurasi model klasifikasi walaupun memerlukan waktu pemrosesan yang relatif lebih lama dibandingkan lainnya, meskipun dalam praktiknya perbedaan ini tidak akan terasa karena semua model memerlukan waktu paling lama 0.002 detik dalam memproses satu baris masukkan.

Kata Kunci: BiLSTM, TF-IDF, TextCNN, ResNet, *logistic regression*, *pre-processing*, SQL *injection, natural language processing*

ABSTRACT

SEMBADA MAJU, Jumeneng Community Database System, is a desktop-based population application created to support population administration activities in Dukuh Jumeneng. This application is not connected to the internet considering the risk of data theft. This research was conducted to reduce this risk by detecting SQL injection using NLP before execution. In this study, four types of classification models and three types of pre-processing will be evaluated in detecting SQL injection. The classification model in question is logistic regression, BiLSTM, TextCNN, and ResNet. Meanwhile, the pre-processing in question is generalization, elimination, and no pre-processing. The purpose was to see the performance of each classification model in detecting SQL injection and the effect of different pre-processing on the performance and processing time of each classification model.

Each combination of classification models with pre-processing methods will be trained with a small dataset containing SQL injection and regular statements. The performance and training time of each combination will be evaluated at this stage. The trained model will then be validated by classifying it on a larger dataset consisting entirely of SQL injection. Validation is carried out to determine whether the model can identify SQL injections that are not included in the training dataset or not. The accuracy and average processing time from pre-processing to classification by each combination will be evaluated at this stage.

The evaluation results show that the generalization pre-processing method tends to help models in achieving the best performance. At the training stage, almost every model achieved ~99% accuracy, except for ResNet with a pre-processing method of elimination. This combination is broken and classified all inputs as SQL injections, thus the accuracy is only ~34%. Validation using larger dataset shows performance decline across all models, with logistic regression's accuracy dropping drastically to ~19%, indicating its inability to generalize. On the other hand, deep learning models show better performance. BiLSTM shows accuracy in the range of 83-91%, while ResNet shows accuracy in the range of 78-92%. In both models, the best performance is achieved by doing generalization as pre-processing. For TextCNN, the performance doesn't change much across different pre-processing, showing accuracy in range of 84-86%.

In conclusion, deep learning models such as BiLSTM, TextCNN, and ResNet generalize better to unforseen data compared to traditional machine learning model like logistic regression. Among pre-processing methods, generalization consistently yields the best accuracy, albeit with slightly higher processing times that remains negligible since at worst, a model only needs 0.002 seconds to process a single string of input.

Keywords**:** BiLSTM, TF-IDF, TextCNN, ResNet, *logistic regression*, *pre-processing*, SQL *injection, natural language processin*