PLAGIARISM CHECK.ORG

2.29%

SIMILARITY OVERALL

SCANNED ON: 21 JUL 2025, 10:07 AM

Similarity report

Your text is highlighted according to the matched content in the results above.

CHANGED TEXT
2.28%

Report #27589607

4 1 BAB I PENDAHULUAN 1.1 Latar Belakang Masalah Perkembangan teknologi informasi yang pesat telah membawa kemudahan dalam berbagai aspek kehidupan, termasuk dalam pertukaran data dan komunikasi. Namun, kemajuan ini juga dimanfaatkan oleh pihak-pihak yang tidak bertanggung jawab untuk melakukan kejahatan siber. Salah satu metode yang sering digunakan adalah penyebaran URL berbahaya, yaitu tautan yang dirancang untuk mengelabui pengguna agar mengakses situs berbahaya yang dapat mencuri data pribadi atau menginfeksi perangkat dengan malware. Menurut Rafi (2023), serangan umumnya melibatkan teknik phishing dan rekayasa sosial, dimana pelaku membuat domain yang menyerupai situs resmi. Pengguna yang kurang waspada dapat diarahkan ke situs palsu yang tidak hanya mencuri informasi sensitif seperti kredensial akun media sosial atau perbankan, tetapi juga menyebarkan malware ke seluruh jaringan perangkat korban. Dampak dari serangan ini sangat merugikan, mulai dari kerusakan sistem hingga hilangnya data penting yang sulit dipulihkan. Upaya pencegahan tradisional, seperti pelatihan kesadaran keamanan siber, pemblokiran pop-up, dan penggunaan perangkat lunak antivirus, telah diterapkan untuk mengurangi risiko serangan. Namun, pendekatan ini bersifat reaktif dan memerlukan intervensi manual, sehingga kurang efektif dalam menghadapi serangan yang semakin kompleks dan dinamis. Selain itu, banyak perangkat lunak keamanan

yang tidak mampu mendeteksi URL berbahaya baru tanpa pembaruan yang terus-menerus. 2 Seiring dengan perkembangan teknologi kecerdasan buatan, pendekatan berbasis machine learning telah digunakan untuk meningkatkan deteksi ancaman siber. Namun, banyak model yang masih memiliki keterbatasan dalam mengenali pola teks yang kompleks dan konteks yang berubah-ubah. Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan integrasi Large Language Model (LLM) berbasis BERT ke dalam sistem Security Information and Event Management (SIEM). LLM memiliki kemampuan dalam memahami dan menganalisis pola bahasa alami secara mendalam, sehingga dapat meningkatkan akurasi dalam mendeteksi URL berbahaya secara otomatis dan real- time. Dalam implementasinya, dataset yang terdiri dari URL aman dan berbahaya akan digunakan untuk melatih model BERT. Model yang telah dilatih kemudian diintegrasikan ke dalam sistem SIEM, yang memungkinkan deteksi dini terhadap akses ke URL berbahaya dan memberikan respons aktif untuk mencegah potensi serangan. Penelitian ini menggunakan SIEM berbasis Wazuh, yang dikenal dapat memantau keamanan, mendeteksi ancaman, dan menanggapi insiden secara real- time. Wazuh merupakan SIEM berbasis open-source yang memungkinkan pengguna untuk mengembangkan sistem sesuai dengan kebutuhan yang diperlukan, sehingga pengembangan pada penelitian ini menjadi optimal. Dengan pendekatan ini, sistem keamanan siber

diharapkan dapat beroperasi secara proaktif dan adaptif terhadap ancaman yang terus berkembang. Tujuan dari penelitian ini adalah merancang dan mengimplementasikan SIEM berbasis BERT yang mampu mendeteksi dan memblokir URL berbahaya dengan akurasi tinggi dan respons yang cepat, sehingga meningkatkan perlindungan terhadap ancaman siber yang semakin kompleks. 3 1.2 Identifikasi Masalah Meningkatnya jumlah serangan siber di era digital menjadi ancaman serius bagi keamanan data pengguna. Salah satu metode serangan yang paling umum adalah melalui URL berbahaya, yaitu tautan yang secara sengaja dibuat untuk mengarahkan pengguna ke situs berbahaya. Situs ini dapat digunakan untuk mencuri informasi pribadi, menginfeksi perangkat dengan malware, hingga mengakses sistem tanpa izin. Untuk itu, dibutuhkan pendekatan yang lebih adaptif dan cerdas melalui pemanfaatan teknologi kecerdasan buatan, khususnya Large Language Model (LLM) berbasis BERT. Integrasi BERT ke dalam sistem Security Information and Event Management (SIEM) diharapkan dapat menjadi solusi proaktif dalam mendeteksi dan mencegah akses ke URL berbahaya secara otomatis dan real-time. 1.2.1 Rumusan masalah Berdasarkan identifikasi masalah yang ditemukan, penelitian ini merumuskan masalah yaitu, “Bagaimana mengembangkan Security Information & Event Management (SIEM) dengan Large Language Model berbasis BERT untuk mendeteksi, memprediksi dan mencegah akses pada URL berbahaya? 1.2.2 Batasan masalah Batasan masalah pada penelitian ini: a) Data URL berbahaya yang digunakan dalam pengujian pada Security Information & Event Management (SIEM) harus diperbarui secara berkala, seiring terus bermunculannya URL berbahaya yang belum diketahui. b) Security Information & Event Management (SIEM) ini tidak bertujuan untuk menggantikan solusi keamanan yang sudah ada, tetapi sebagai tambahan lapisan keamanan tambahan untuk pengguna internet dalam lingkup jaringan lokal. 4 1.3 Tujuan Penelitian Membangun dan mengimplementasikan layanan deteksi URL berbahaya menggunakan Large Language Model (LLM) berbasis BERT, mengintegrasikan

seluruh sistem deteksi dan pemblokiran dengan Wazuh SIEM untuk monitoring, alerting, pelaporan insiden keamanan, meminimalisir potensi akses pada phishing, malware, dan ancaman berbasis URL lainnya serta menyediakan alert log terpusat pada SIEM untuk analisis dan mitigasi insiden. 1.4 Manfaat Penelitian Penelitian ini memiliki manfaat bagi tiga pihak: masyarakat, peneliti, dan ilmu pengetahuan. Berikut adalah rangkuman dan penjelasan manfaat tersebut. a) Manfaat bagi Masyarakat Solusi otomatis yang tersedia untuk memperkuat keamanan jaringan lokal—terutama di pemerintahan, sekolah, dan usaha kecil menengah—dengan memblokir URL berbahaya sebelum dapat diakses pengguna merupakan manfaat praktis dari penelitian ini. Dengan menggabungkan BERT ke dalam SIEM, risiko pencurian data pribadi, gangguan layanan, dan penyebaran malware dapat diminimalisir. Sebagai hasilnya, penerapan sistem ini diharapkan menurunkan tingkat insiden siber, meningkatkan kepercayaan pengguna terhadap layanan online, dan meningkatkan kesadaran proaktif masyarakat tentang keamanan informasi. b) Manfaat bagi Peneliti Salah satu manfaat bagi peneliti dari penelitian ini adalah meningkatkan kemampuan bidang informatika dalam bidang keamanan siber dengan menerapkan dan menerapkan metode machine learning untuk mendeteksi URL berbahaya yang berbahaya pada jaringan. c) Manfaat bagi Ilmu Pengetahuan Salah satu manfaat bagi ilmu pengetahuan dalam penelitian ini, yaitu melakukan implementasi Security Information & Event Management 5 (SIEM) untuk deteksi URL berbahaya pada koneksi jaringan yang berbahaya dan dapat mencegah adanya akses yang dapat membahayakan data pengguna pada jaringan. Peneliti juga berharap penelitian ini dapat menjadi referensi untuk penelitian berikutnya.
1.5 Kebaruan Penelitian ini memberikan kebaruan berupa integrasi Large Language Model (LLM) berbasis BERT untuk meningkatkan akurasi pengolahan data dan melatih model dalam mengidentifikasi URL berbahaya, serta implementasi Security Information & Event Management (SIEM) yang memanfaatkan BERT dalam mendeteksi dan memblokir URL

berbahaya secara otomatis pada jaringan lokal (LAN). 1.6 Kerangka Penulisan Laporan ini disusun berdasarkan pedoman yang sudah ditetapkan oleh Lembaga Penjamin Mutu Universitas Pembangunan Jaya, dan sesuai dengan sistematika dalam Program Studi Informatika, yang terdiri dari 6 bab.; BAB I PENDAHULUAN Bab ini terdiri dari subab latar belakang dilakukannya penelitian, identifikasi masalah yang berupa rumusan masalah, tujuan penelitian, manfaat penelitian bagi masyarakat, peneliti, dan ilmu pengetahuan, kebaruan, dan kerangka penulisan. BAB II TINJAUAN PUSTAKA Bab ini terdiri dari sub bab pencapain terdahulu dan tinjauan teoritis yang mendukung penelitian. BAB III TAHAPAN PELAKSANAAN 6 Bab ini menjelaskan tahapan yang diperlukan untuk menjalankan penelitian dari awal hingga akhir, dan juga akan menjelaskan metode penelitian yang dipilih. BAB IV PERANCANGAN Bab ini menjelaskan tahapan yang dilakukan untuk melakukan pengembangan pada penelitian ini, berisi bagaimana sistem dan metode yang digunakan untuk mencapai hasil akhir. BAB V HASIL DAN PEMBAHASAN Bab ini akan membahas hasil dan temuan penelitian dan memberikan penjelasan tentang temuan dalan sistem yang dikembangkan. BAB VI KESIMPULAN Bab ini akan memuat hasil penelitian yang disusun secara sistematis tetapi juga memberikan rekomendasi untuk pengembangan selanjutnya. . 7 BAB II TINJAUAN PUSTAKA 2.1 Pencapaian Terdahulu Penelitian ini bertujuan untuk mengembangkan Security Information & Event Management (SIEM) untuk mencegah akses ke URL berbahaya dengan menggunakan algoritma Large Language Model. Untuk mendukung penelitian ini, berbagai pencapaian terdahulu telah ditinjau untuk memahami pendekatan dan metode yang telah digunakan untuk mendeteksi dan mencegah akses pada URL berbahaya. Beberapa penelitian yang relevan adalah sebagai berikut: Tabel 2.1 Penelusuran Literatur Penelitian ke-1 Nama Penulis Nguyen et al. (2020) Judul Detecting abnormal DNS traffic using unsupervised machine learning Hasil Membandingkan kinerja empat algoritma pembelajaran mesin yang tidak

diawasi: K-means, Gaussian Mixture Model (GMM), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), dan Local Outlier Factor (LOF) pada Boss dari SOC Dataset Versi 1 (Botsv1) dataset dari proyek Splunk untuk mendeteksi Malicious DNS Traffic. Penelitian ke-2 Nama Penulis Rafi et al. (2023) Judul Analisis Malicious URL pada file menggunakan metode K- Means Clustering berbasis Host-Based Feature Extraction Hasil Menghasilkan sebuah dataset URL yang memiliki fitur DNS Record dari URL yang akan digunakan sebagai data untuk 8 melakukan clustering dengan K-Mean. Dengan menggunakan nilai k=2, kluster benign dan malicious URL dihasilkan sebagai visualisasi dari data hasil clustering dengan K-Mean. Penelitian ke-3 Nama Penulis (Do Xuan et al., 2020) Judul Malicious URL Detection based on Machine learning Hasil/Bukti Melakukan deteksi Malicious URL menggunakan algoritma Random Forest dan Support Vector Machine untuk memprediksi adanya Malicious URL pada sistem dan dikelompokkan berdasarkan tingkat prediksi URL yang aman atau yang tidak aman. Penelitian ke-4 Nama Penulis (Singh & Roy, 2020) Judul Detecting Malicious DNS over HTTPS Traffic Using Machine learning Hasil/Bukti Analisa terhadap deteksi Malicious DNS pada trafik jaringan menggunakan machine learning dengan algoritma Random Forest. Dimana data diklasifikasikan menjadi DNS yang aman dan DNS yang tidak aman berdasarkan hasil confusion matrix dari dataset yang diambil. Penelitian ke-5 Nama Penulis (Tiwari, 2019) Judul Suspicious URL Detection using Dynamic Learning Model with Machine learning Hasil/ Bukti Pembuatan aplikasi sederhana berbasis python untuk mendeteksi adanya URL yang mencurigakan dengan 9 menggunakan algoritma Linear SVM Classifier, K Nearest Neighbors Classifier, Random Forest Classifier dan menunjukkan hasil akhir dalam bentuk persentase terhadap URL tujuan yang dicari. Penelitian ke-6 Nama Penulis (Hilal et al., 2023) Judul Malicious URL Classification Using Artificial Fish Swarm Optimization and Deep Learning Hasil Mengembangkan model

Artificial Fish Swarm Algorithm dengan Deteksi dan Klasifikasi URL Berbahaya dengan menggunakan Deep Learning (AFSADL-MURLC). Model AFSADL-MURLC yang disajikan bertujuan untuk membedakan URL berbahaya dari URL asli. Hasil simulasi menegaskan keunggulan model AFSADL-MURLC yang diusulkan dibandingkan dengan pendekatan terbaru berdasarkan berbagai ukuran. Penelitian ke-7 Nama Penulis (Sim & Kim, 2023a) Judul Malicious URL Detection based on Supervised and Unsupervised Learning using MobileBERT Embedding Hasil/Bukti Kinerja yang lebih tinggi dicapai ketika dimensi vektor embedding dikurangi menggunakan PCA daripada autoencoder, dan URL berbahaya dapat dideteksi dengan probabilitas tinggi hanya dengan menggunakan vektor yang di-embedding melalui MobileBERT tanpa pengurangan dimensi. Dalam Unsupervised Learning, recall keseluruhan lebih tinggi daripada presisi, dan peningkatan jumlah sampel data normal meningkatkan kinerja deteksi. 10 Penelitian ke-8 Nama Penulis (Shaheed & Kurdy, 2022) Judul Web Application Firewall Using Machine learning and Features Engineering Hasil Mengembangkan model firewall aplikasi web yang menggunakan Feature Extracting menggunakan Dataset CSIC 2010, HTTPParams 2015, Hybrid dataset (CSIC 2010 and HTTPParams), machine learning Naïve Bayes, Logistc Regression, Decision Tree, Support Vector Machine. dan rekayasa fitur untuk mendeteksi serangan web umum. Model yang diusulkan menganalisis permintaan yang masuk ke server web, mem-parsing permintaan tersebut untuk mengekstrak empat fitur yang menggambarkan bagian permintaan HTTP (URL, payload, dan header), dan mengklasifikasikan apakah permintaan tersebut normal atau anomali. Hasilnya menunjukkan bahwa model yang diusulkan mencapai akurasi klasifikasi sebesar 99,6% dengan dataset yang digunakan dalam penelitian ini dan 98,8% dengan dataset dari server web nyata. Penelitian ke-9 Nama Penulis (Muzaki et al., 2020) Judul Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall Hasil/Bukti Melakukan pengujian,

pemantauan dan pemblokiran menggunakan ModSecurity sebagai WAF dengan metode Reverse Proxy untuk mencegah serangan Cross-site scripting, SQL Injection dan web vulnerability scanning yang tidak sah. Dari hasil yang dilakukan, seperti cross-site scripting, SQL 11 injection, dan pemindaian kerentanan web yang tidak sah, semua ancaman berhasil digagalkan oleh ModSecurity dan metode reverse proxy yang diimplementasikan dalam WAF. Penelitian ke-10 Nama Penulis Almasri, M. N., et al. (2024) Judul Detecting Phishing URLs using the BERT Transformer Model Hasil/Bukti Encoder yang diusulkan mengungguli model deep learning berbasis karakter state-of-the-art dan model BERT yang berfokus pada keamanan siber di berbagai tugas dan dataset. Klasifikasi yang dihasilkan mencapai akurasi 95-99% dalam mendeteksi situs berbahaya dari URL mereka dengan false positive yang sederhana. Penelitian ke-11 Nama Penulis Li, Y., et al. (2024) Judul Continuous Multi-Task Pre-training for Malicious URL Detection and Webpage Classification Hasil/Bukti urlBERT mengungguli model yang dilatih awal standar dan mode multi-task-nya mampu memenuhi kebutuhan dunia nyata. Menunjukkan potensi untuk deteksi URL berbahaya dan klasifikasi halaman web secara bersamaan. 3 12 2.2 Tinjauan Teoritis Dalam konteks penelitian ini, tinjauan teori digunakan sebagai dasar untuk penyesuaian dengan topik penelitian dan sebagai pedoman untuk melakukan penelitian yang baik. 3 2.2 1 URL Menurut Do Xuan et al (2020), Uniform Resource Locator (URL) digunakan untuk merujuk pada sumber daya di Internet. Karakteristik dari URL adalah dua komponen dasar seperti ID protokol, yang menunjukkan protokol mana yang harus digunakan, dan nama sumber daya, yang menentukan alamat IP atau nama domain di mana sumber daya berada. Anda dapat melihat bahwa setiap URL memiliki struktur dan format yang spesifik. 2.2.2 Security Information & Event Management (SIEM) Menurut Bezas & Filippidou (2023) dan Horng et al., (2023), Security Information & Event Management (SIEM) adalah alat keamanan siber yang menggabungkan

Security Information Management (SIM) dan Security Event Management (SEM) untuk menawarkan pendekatan komprehensif untuk deteksi dan respons ancaman siber. Sistem SIEM mengumpulkan, menganalisis, menormalkan, dan mengkorelasikan data dari berbagai sumber untuk mengidentifikasi potensi ancaman siber secara langsung dan menawarkan pandangan terpusat tentang posisi keamanan organisasi. 2.2.2.1 Wazuh Wazuh adalah platform keamanan siber open-source yang terkenal dan lengkap yang dapat memantau keamanan, mendeteksi ancaman, dan menanggapi insiden secara real-time. Ini berfungsi sebagai solusi yang kuat untuk sistem deteksi intrusi berbasis host (HIDS) dan manajemen informasi dan peristiwa keamanan (SIEM). Ini memungkinkan organisasi untuk mengumpulkan, menganalisis, dan mengkorelasikan data keamanan dari berbagai sumber, seperti log, kejadian, dan lalu lintas jaringan.
Wazuh beroperasi dalam arsitektur client-server. Agent Wazuh 13 dipasang pada endpoint atau perangkat yang dipantau, seperti laptop, server, atau kontainer. Tugasnya adalah mengumpulkan data sistem, log, dan informasi keamanan lainnya. Selanjutnya, data dikirimkan ke Manajer Wazuh Pusat untuk dianalisis, diindeks, dan disimpan. Manajer Wazuh menganalisis data sesuai dengan beberapa aturan yang telah dikonfigurasi, dan akan memberikan peringatan (pesan) ketika ada kejadian yang sesuai dengan aturan tertentu. Wazuh Dashboard yang terintegrasi kemudian memungkinkan visualisasi data yang dianalisis.
2.2.3 Large Language Model Large Language Models (LLM) adalah jenis model pembelajaran mesin berskala besar yang dirancang untuk memproses dan memahami bahasa alami. Model ini dilatih dengan jumlah data yang besar menggunakan metode self- supervised learning untuk membentuk representasi umum dari bahasa. Bommasani et al. (2022) menyebut model seperti ini sebagai foundation models, karena dapat diterapkan pada berbagai tugas tanpa harus dilatih ulang secara spesifik untuk setiap tugas. Selain itu, Zhao et al. (2023) menyatakan bahwa LLM digunakan untuk menangkap general-purpose

linguistic representations, sehingga dapat dimanfaatkan dalam berbagai aplikasi NLP seperti, question answering, text summarization, hingga code generation. LLM adalah model probabilistik yang dibangun di atas jaringan saraf dengan miliaran atau bahkan triliunan parameter. Parameter ini berfungsi sebagai penyimpan pengetahuan linguistik dan faktual yang diekstraksi dari data pelatihan. Skala yang masif inilah yang menjadi kunci dari kemampuan LLM untuk melakukan penalaran, generalisasi, dan pemahaman bahasa yang kompleks. Berikut adalah cara kerja dari LLM; Arsitektur Basis: Transformer Menurut Vaswani et al. (2017), merupakan dasar dari arsitektur Large Language Models (LLM) kontemporer. Mekanisme perhatian diri, atau self-attention mechanism, adalah inovasi utama dalam arsitektur ini. Dengan bantuan mekanisme ini, model dapat menimbang dan mengevaluasi relevansi kontekstual dari setiap token dalam sekuens data secara bersamaan dengan 14 semua token lainnya. Kemampuan ini memungkinkan pemahaman konteks yang lebih akurat, mengatasi keterbatasan model sekuensial sebelumnya dalam menangani dependensi jangka panjang. Pelatihan Dua Tahap Pengembangan LLM Pre-training (Pelatihan Awal): Tahap pertama adalah pelatihan pra-latihan yang dilakukan sendiri. Pada tahap ini, model dilatih pada korpus teks yang sangat besar untuk tujuan pemodelan bahasa (bahasa modeling), yang mencakup hal-hal seperti memprediksi token yang akan datang. Metode ini memungkinkan model menginternalisasi representasi linguistik yang kaya, termasuk pengetahuan faktual, kemampuan penalaran dasar, dan tata bahasa. Fine-tuning, atau penyelarasan: Tujuan dari tahap kedua penyelarasan adalah untuk menyelaraskan (align) perilaku model dengan instruksi dan preferensi manusia. Reinforcement Learning from Human Feedback (RLHF) adalah salah satu metodologi yang paling efektif untuk tujuan ini. Menurut Ouyang et al. (2022), proses RLHF menggunakan umpan balik manusia untuk melatih sebuah reward model, yang kemudian memandu LLM agar menghasilkan respons yang lebih

bermanfaat, jujur, dan aman. 2.2 2 3.1 BERT BERT, singkatan dari Bidirectional Encoder Representations from Transformers, adalah model bahasa inovatif yang diluncurkan oleh Google pada tahun 2018. Jika dibandingkan dengan model bahasa sebelumnya, model ini jauh lebih baik dalam memahami konteks bidireksional kata-kata dalam kalimat. Menurut Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019) BERT dibuat untuk mengajarkan representasi bahasa umum (replikasi bahasa umum) dari korpus teks besar, seperti BookCorpus dan Wikipedia, yang kemudian dapat disesuaikan dengan mudah untuk berbagai tugas pemrosesan bahasa alami (NLP) tanpa melakukan modifikasi arsitektur yang signifikan. 15 Arsitektur Encoder Transformer: Arsitektur encoder model Transformer adalah inti dari BERT. Menurut Vaswani et al. (2017), BERT menggunakan mekanisme self-attention untuk memproses seluruh urutan masukan secara paralel. Ini memungkinkan model untuk menangkap dependensi jarak jauh antar kata dengan sangat efisien dibandingkan dengan model recurrent (RNN) yang memproses sekuensial. Bidireksionalitas Penuh: Berbeda dengan model bahasa sebelumnya, seperti GPT yang hanya unidireksional atau ELMo yang menggabungkan dua arah terpisah, BERT dilatih untuk memahami konteks kata berdasarkan kata di sebelah kiri dan kanannya sekaligus. Ini dicapai melalui dua tugas pra-pelatihan: - Model bahasa yang disembunyikan (MLM): Sebagian kata disembunyikan (masked) dalam input, dan model BERT dilatih untuk memprediksi kata-kata yang disembunyikan tersebut berdasarkan konteks sekitarnya. Dengan tugas ini, model dapat memperoleh pemahaman bidireksional yang kaya tentang bahasa. - Model Next Sentence Prediction (NSP) berfungsi untuk menentukan apakah dua bagian berurutan dari teks adalah pasangan kalimat logis (yaitu, apakah kalimat pertama benar-benar mengikuti kalimat kedua). BERT memperoleh pemahaman yang lebih baik tentang hubungan antar kalimat dari tugas ini, penting untuk tugas seperti menjawab pertanyaan dan inferensi bahasa. Pra-pelatihan (Pre-training) dan Penyesuaian (Fine-tuning): Dua

tahap utama dilakukan dalam model BERT: - Tahap Pra-Pelatihan: Tugas MLM dan NSP digunakan untuk melatih model pada dataset teks yang sangat besar. Pada tahap ini, model belajar representasi bahasa yang umum dan mendalam. - Tahap Penyesuaian: Hanya memerlukan penyesuaian (fine-tuning) dengan dataset yang jauh lebih kecil untuk tugas NLP khusus model BERT yang sudah dilatih sebelumnya. 8 Ini secara signifikan mengurangi jumlah waktu dan sumber daya komputasi yang diperlukan untuk melatih model sejak awal. 16 Representasi Kontekstual: BERT dapat menghasilkan embedding (representasi numerik) kata yang kontekstual berkat bidireksionalitas dan arsitektur Transformer. Representasi kata-kata yang berbeda bergantung pada konteks kalimat, kata yang sama dengan makna yang sama akan memiliki representasi yang berbeda dalam beberapa kalimat. Mekanisme kerja utama dari Bidirectional Encoder Representations from Transformers (BERT) adalah metode pelatihannya, yang dimaksudkan untuk membuat representasi kontekstual dua arah (bidirectional). BERT, yang dilakukan oleh Devlin et al. (2018), memproses seluruh sekuens kata secara bersamaan melalui arsitektur encoder Transformer. Ini membedakannya dari model sebelumnya yang memproses teks secara sekuensial (baik dari kiri ke kanan maupun dari kanan ke kiri). 2 7 Metode ini dicapai melalui dua tugas pre-training yaitu, yaitu Masked Language Model (MLM) dan Next Sentence Prediction (NSP). Tugas pre-training MLM adalah inovasi utama BERT, yang bertujuan untuk mengatasi keterbatasan model bahasa searah (unidirectional). Dalam praktiknya, sekitar lima belas persen dari token yang ada dalam sekuens masukan dipilih secara acak untuk "ditutupi" atau disembunyikan. Sebuah token khusus, [MASK], digunakan untuk menggantikan token-token yang telah dipilih sebelumnya. Menurut Devlin et al. (2018), tujuan model adalah untuk memprediksi token asli yang disembunyikan berdasarkan konteks kedua arah, yaitu token yang mendahului dan yang mengikutinya. Untuk ilustrasi, token "dilatih" dapat disembunyikan sebagai "Model ini [MASK] pada data besar dalam

kalimat "Model ini dilatih pada data besar . Oleh karena itu, untuk memprediksi token asli, model harus mempertimbangkan konteks "Model ini" dan "pada data besar" secara bersamaan. Tujuan matematis dari MLM adalah untuk meminimalkan fungsi kerugian, atau fungsi kerugian, antara distribusi probabilitas prediksi model dan token asli yang disembunyikan. Fungsi kerugian ini biasanya disebut sebagai kerugian cross- entropy. Ini dapat digambarkan sebagai upaya untuk meningkatkan kemungkinan log kata yang benar dalam seluruh konteksnya: 17 Dimana D adalah korpus data, wmasked adalah himpunan token yang disembunyikan dalam sekuens x, dan x \masked adalah token yang tidak disembunyikan dalam sekuens yang sama. BERT juga dilatih dengan tugas Next Sentence Prediction (NSP) untuk membantu model memahami hubungan antar kalimat, yang penting untuk tugas seperti Jawaban Pertanyaan (QA) dan Natural Language Inference (NLI). Sepasang kalimat (A dan B) diberikan kepada model dalam tugas ini, dan mereka kemudian diminta untuk melakukan klasifikasi biner untuk menentukan apakah kalimat B merupakan kelanjutan logis dari kalimat A dalam korpus asli (Devlin et al., 2018). Selama pelatihan, setengah dari pasangan kalimat yang diberikan adalah pasangan yang berurutan, dan setengah lagi adalah pasangan di mana kalimat B dipilih secara acak dari korpus. Untuk klasifikasi ini, representasi output dari token khusus [CLS] yang ditambahkan di awal setiap sekuens input digunakan sebagai representasi gabungan dari seluruh pasangan kalimat. Setelah melalui prosedur diatas, BERT berhasil membuat representasi vektor yang kaya akan informasi kontekstual pada level kata dan kalimat dengan menggabungkan dua tugas pre-training tersebut. Dengan menambahkan hanya satu lapisan output tambahan yang khusus untuk tugas hilir, presentasi ini kemudian dapat disesuaikan dan disesuaikan secara efektif untuk berbagai tugas di bawahnya. 6 18 BAB III TAHAPAN PELAKSANAAN 3.1 Langkah-langkah Pelaksanaan Proses pelaksanaan ini akan membahas proses penyelesaian tugas akhir. Ini mencakup prosedur

pelaksanaan dan teknik pengujian yang digunakan. Gambar 3.1 Langkah-Langkah Perancangan 19 Berikut ini adalah penjelasan dari Gambar diatas: a. Identifikasi Masalah: Tahap yang berisi apa masalah yang ingin dibahas, mencari permasalahan yang relevan, menentukan tujuan penelitian dan memberikan solusi yang sesuai dengan masalah yang diteliti. b. Studi Literatur: Tahap studi literatur dilakukan untuk mencari penelitian terdahulu yang relevan pada penelitian ini, mulai dari penelusuran jurnal, pemilahan jurnal dan sitasi terhadap jurnal yang memiliki latar belakang atau judul yang relevan. c. Pengumpulan Data: Proses pengumpulan data terdiri dari pencarian sumber data dan pra-pemrosesan data. Pencarian sumber data yaitu mengumpulkan dataset yang diperlukan pada penelitian ini, yakni adalah kumpulan dataset URL yang diketahui aman dan berbahaya. Pra-pemrosesan data dilakukan setelah selesai mengumpulkan dataset yang diperlukan, kemudian dataset yang dikumpulkan akan dipilah, diekstrasi fitur dan data dinormalisasi. d. Pengembangan Model : Pengembangan model yang dilakukan adalah, merancang algoritma Large Language Model yang akan digunakan untuk mendeteksi Malicious URL dengan bahasa pemrograman Python dan melakukan pelatihan/training menggunakan dataset yang sudah dikumpulkan. e. Evaluasi Model: Evaluasi pada pengujian kinerja pada algoritma Large Language Model yaitu; akurasi, presisi dan analisis hasil pengujian untuk menilai efektivitas algoritma-nya dalam melakukan deteksi URL berbahaya. 20 f. Pengembangan Security Information & Event Management (SIEM): Pengembangan Security Information & Event Management (SIEM) akan dimulai dari mengintegrasikan model machine learning Large Language Model yang sudah dilatih untuk melakukan deteksi URL berbahaya dan diuji untuk melakukan monitoring pada sistem yang sudah diintegrasikan dengan algoritma machine learning Large Language Model. g. Pengujian Security Information & Event Management (SIEM): Pengujian Security Information & Event Management (SIEM) dilakukan dalam skala bertahap, dimulai dari lingkup

menggunakan mesin virtual dan komputer pribadi, hingga pengujian pada skala yang lebih besar. h. Pembuatan Laporan: Segala aktivitas yang berhubungan pengujian aplikasi, hasil yang dicapai, analisis hingga implementasi aplikasi akan ditulis pada laporan akhir. 21 3.2 Tahap Penerapan Algoritma Penerapan Algoritma pada penelitian ini menggunakan algoritma Large Language Model untuk melakukan deteksi URL yang berbahaya dengan cara mengelompokkannya berdasarkan karakteristik tertentu. Berikut ini adalah Gambar proses beserta penjelasan untuk mencapai fungsionalitas deteksi URL yang berbahaya: a) Pengumpulan & Persiapan Data: Langkah pertama adalah mengumpulkan kumpulan data yang berisi daftar URL dan label yang menunjukkan apakah URL tersebut aman (aman) atau malicious/phishing (berbahaya). Setelah data dikumpulkan, dilakukan proses pembersihan, yang meliputi penghapusan protokol HTTP atau https://, penghapusan spasi, dan pengaturan penulisan yang standar. Label menggunakan format numerik, biasanya untuk benign dan 1 untuk malicious. b) Tokenisasi & Encoding: Setelah data selesai, tokenizer, seperti BERT, digunakan untuk mengubah setiap URL menjadi format yang dapat dipahami oleh model Gambar 3.2 Tahap Penerapan Algoritma 22 LLM. Proses tokenisasi memecah URL menjadi potongan kecil, yang dikenal sebagai token, yang kemudian diubah menjadi angka, yang dikenal sebagai input ID. Bersamaan dengan itu, attention mask dibuat untuk menunjukkan input mana yang harus diproses oleh model. c) Konfigurasi Model: Model LLM, seperti BERT untuk klasifikasi, diatur dengan menambahkan lapisan klasifikasi pada bagian akhir. Model ini kemudian disiapkan untuk mengklasifikasikan input menjadi dua kelas (benign atau malicious). Hyperparameter seperti learning rate, batch size, dan jumlah epoch diatur untuk mendukung proses pelatihan yang optimal. d) Fine-Tuning (Pelatihan Model): Model LLM akan dikonfigurasi dan dilatih menggunakan dataset yang telah disiapkan. Tujuan dari proses ini adalah untuk mengubah bobot internal model berdasarkan pola-pola yang ditemukan dalam URL

dan labelnya, sehingga model dapat mengidentifikasi fitur URL yang berbahaya. e) Evaluasi dan Tuning: Setelah pelatihan selesai, model dievaluasi menggunakan data uji untuk mengukur kinerjanya dengan metrik seperti akurasi, ketepatan, recall, dan skor F1. f) Export dan Optimasi Model: Model yang sudah terlatih disimpan (dieksport) dalam bentuk file dan dapat dioptimalkan untuk kebutuhan pada saat monitoring. g) Deploy sebagai API Model yang telah disimpan dijalankan dalam bentuk layanan (API) menggunakan framework Flask. API ini memungkinkan sistem lain untuk mengirimkan URL dan menerima hasil klasifikasi dari model. 23 h) Integrasi ke dalam sistem SIEM Setelah itu, skrip respons aktif digunakan untuk menghubungkan API prediksi LLM ke sistem Wazuh. Ketika Wazuh menemukan URL yang mencurigakan dalam log, ia memanggil API tersebut dan menerima hasil klasifikasi. Jika URL terindikasi berbahaya, Wazuh dapat melakukan aksi otomatis seperti memblokir URL dan alamat IP. i) Perawatan dan Maintenance Setelah sistem berjalan, proses pemantauan dilakukan untuk memastikan model tetap akurat. Ini memungkinkan set data diperbarui secara berkala dengan data terbaru, dan model dilatih ulang untuk menangani jenis serangan baru, yang menghasilkan siklus berkelanjutan untuk menjaga efektivitas deteksi. 24 3.3 Metode Pengembangan Perangkat Lunak Tahap pengembangan aplikasi dibuat berdasarkan Secure Software Development Life Cycle(SSDLC) untuk penelitian ini, dan akan dijelaskan pada gambar serta penjelasan berikut; a) Planning/Perencanaan Tahap awal untuk perancangan, dimulai dari pelatihan algoritma Large Language Model untuk deteksi URL berbahaya menggunakan dataset yang sudah diambil, pembuatan topologi jaringan yang akan digunakan beserta lingkupnya, pengembangan dan implementasi algoritma ke dalam aplikasi web yang akan dibuat. b) Requirements/Analisis Kebutuhan Analisis kebutuhan yang diperlukan adalah pelatihan algoritma Large Language Model untuk deteksi URL berbahaya dari dataset yang diambil, proses pemilahan dataset yang

diuji dan hasil deteksi dari dataset yang duji, merancang topologi jaringan untuk skala pengujian dan merancang aplikasi web yang akan dibuat. c) Design/Desain Tahap desain dilakukan untuk merancang arsitektur pada aplikasi web yang akan dibuat. Mengintegrasikan algoritma Large Language Model sebagai detektor URL berbahaya, menggunakan Security Information & Gambar 3.3 Secure Software Development Life Cycle 25 Event Management (SIEM) Wazuh sebagai aplikasi antarmuka untuk menampilkan hasil deteksi. d) Implementation /Implementasi Implementasi pengembangan Security Information & Event Management (SIEM) akan menggunakan Wazuh dan algoritma Large Language Model yang sudah dilatih. Kemudian model data digunakan untuk menyimpan URL dan hasil analisis yang sudah dilakukan, serta membuat fungsi untuk memproses input URL dan menerapkan algoritma Large Language Model. e) Testing/Pengujian Pengujian aplikasi meliputi; functional testing/uji fungsionalitas untuk memastikan fitur-fitur yang ada bekerja sesuai spesifikasi. Performance testing/uji performa untuk menguji kinerja aplikasi, seperti waktu respons dan penggunaan sumber daya. Security testing/Uji keamanan untuk memastikan tidak ada kerentanan terhadap serangan keamanan. Usability testing/Uji kemudahan penggunaan aplikasi untuk memastikan antarmuka sudah sesuai kebutuhan dan mudah digunakan. f) Deployment/Penerapan Tahap penerapan akan dilakukan pada lingkungan, lingkup dan topologi jaringan yang sudah ditentukan, serta melakukan pemantauan untuk memastikan kinerja dan mendeteksi aktivitas yang mencurigakan. g) Maintenance/Pemeliharaan Pemeliharaan sistem diperlukan untuk memastikan pembaharuan keamanan tetap terjaga untuk menangkal ancaman, pemeliharaan secara berkala untuk memastikan kinerja tetap optimal dan respon insiden keamanan untuk melakukan pemulihan jika ada terjadi penyerangan. 26 h) Review /Tinjauan Tinjauan dilakukan secara berkala, untuk memastikan agar semua prosedur dapat diikuti dan lebih mudah untuk melakukan perbaikan jika ada masalah. 3.4 Metode Pengujian Untuk memastikan

kinerja sistem sudah optimal, akan dilakukan pengujian sistem aplikasi web untuk mencegah akses URL berbahaya menggunakan algoritma Large Language Model menggunakan dua metode pengujian; pertama, pembuatan skenario topologi jaringan dan pengembangan Security Information & Event Management (SIEM) yakni yang digunakan adalah Wazuh. Berikut adalah penjelasan tentang metode pengujian yang digunakan. 3.4.1 Pengujian Skenario Topologi Jaringan Pengujian skenario topologi jaringan dilakukan secara real-time. Pada topologi jaringan yang digunakan, prerangkat terdiri dari laptop/komputer fisik dengan perangkat lunak VirtualBox, yang berisi mesin virtual Server dengan Security Information & Event Management (SIEM) dan Client fisik berupa komputer/laptop. Gambar 3.4 Topologi Jaringan Pengujian 27 3.4.2 Pengujian SIEM Pengujian Security Information & Event Management (SIEM) dilakukan menggunakan metode white box dan metode black box, untuk memastikan tidak ada kendala fungsionalitas. 3.4.2.1 White Box Metode White Box diterapkan untuk memeriksa dan menguji logika internal serta struktur kode algoritma yang membentuk sistem deteksi dan pencegahan akses URL berbahaya. Pengujian ini dilakukan dengan menganalisis kode dari model LLM, API Flask yang digunakan untuk menerima dan memproses URL, serta skrip Active Response yang ada di Wazuh. Dengan metode ini, penguji dapat memastikan bahwa setiap fungsi dalam proses pelatihan model, prediksi URL, hingga pemicu tindakan pemblokiran oleh Wazuh berjalan sesuai dengan rancangan. Selain itu, pengujian White Box juga memungkinkan identifikasi kesalahan logika, bug, atau alur komunikasi yang tidak efisien antara komponen-komponen sistem, termasuk integrasi antara Wazuh dan model deteksi berbasis LLM. 3.4.2.2 Black Box Metode Black Box dilakukan dengan fokus pada evaluasi fungsionalitas sistem secara menyeluruh dari perspektif pengguna eksternal tanpa melihat struktur internal sistem. Pengujian dilakukan dengan cara mengirimkan berbagai URL secara langsung dari sisi client, baik URL benign, phishing,

maupun malicious, untuk mengamati bagaimana sistem merespons input tersebut. Tujuan dari pengujian ini untuk memastikan bahwa sistem mampu mendeteksi ancaman dengan benar, mengklasifikasikannya melalui LLM, serta menjalankan Active Response secara otomatis jika URL terdeteksi sebagai berbahaya. Pengujian ini juga mencakup pengujian keamanan dan performa, seperti bagaimana sistem menangani akses dalam kondisi lalu lintas jaringan yang tinggi atau ketika menerima input URL yang tidak biasa (misalnya dengan encoding atau domain baru). Hasil dari pengujian ini digunakan untuk menilai apakah sistem telah memenuhi kebutuhan fungsional dan keamanan dari sisi pengguna. 28 BAB IV PERANCANGAN 4.1 Analisis Sistem Terdahulu Pencegahan akses terhadap URL berbahaya merupakan salah satu upaya dalam lingkup keamanan siber, untuk memastikan pengguna dalam jaringan tetap terjaga dari serangan seperti phishing, malware, ransomware dan sebagainya. Pendekatan yang umum dilakukan adalah memantau sistem SIEM secara manual, serta berpotensi melewatkan setiap aktivitas log pada jaringan yang terjadi dalam jumlah banyak dan dalam waktu yang cepat. Dampak yang terjadi pada pendekatan umum ini, selain dapat melewatkan informasi log dalam kurun waktu yang cepat, berpotensi tidak dapat memantau log dengan efektif, serta memerlukan waktu lebih untuk melihat dan menganalisa log yang sudah terlewat satu per satu. Model LLM merupakan salah satu model dalam machine learning yang berfungsi untuk memproses, memahami serta mengidentifikasi pola dalam bentuk teks, yakni dalam penelitian ini adalah mengidentifikasi pola penulisan URL. Pendekatan dengan model LLM berbasis BERT pada penelitian ini, selain dapat memprediksi potensi adanya URL yang berbahaya namun belum diketahui, dapat membantu mencegah serta mendeteksi adanya URL berbahaya secara otomatis. Pada penelitian yang dilakukan oleh Chang, et al (2022) dan Elsadig, et al (2022), model LLM berbasis BERT terbukti dengan memproses URL untuk mengelompokkan secara efektif dan melatih model BERT untuk

mengklasifikasikan karakteristik string pendek, metode ini mencapai akurasi 98,30%, tingkat penarikan 95,21%, dan nilai F1 94,33%. 29 4.2 Spesifikasi Kebutuhan Sistem Baru Untuk memastikan sistem berjalan dengan baik, diperlukan spesifikasi yang sesuai dengan rekomendasi dari sistem yang dijalankan. Pelatihan model Large Language Model dilakukan untuk mempelajari dan memprediksi potensi dari URL yang berbahaya, pelatihan model ini menggunakan dataset. 10 Spesifikasi ini terdiri dari perangkat keras (hardware) dan perangkat lunak (software). 4.2 1 1 Spesifikasi Kebutuhan Perangkat Lunak Berikut adalah komponen perangkat lunak yang digunakan dalam penelitian ini, beserta dengan rincian yang diperlukan. Tabel 4.1 Tabel Kebutuhan Perangkat Lunak No. Perangkat Lunak Deskripsi 1. Visual Studio Code Aplikasi untuk merancang kode pemrograman dan melatih model Large Language Model. 2. Virtual Box Aplikasi untuk menjalankan sistem SIEM dalam bentuk mesin virtual. 3. ISO Wazuh Sistem operasi SIEM yang digunakan untuk sistem deteksi, blokir dan logging. 4. Web Browser Aplikasi untuk menjelajahi situs web untuk mencari informasi. 5. Python Bahasa pemrograman yang digunakan untuk melatih model. 4.2 1 2 Spesifikasi Kebutuhan Perangkat Keras Berikut adalah komponen perangkat keras yang digunakan dalam penelitian ini, beserta dengan spesifikasi dari setiap perangkat. 30 Tabel 4.2 Tabel Kebutuhan Perangkat Keras No. Nama Perangkat Keras Spesifikasi 1. Processor AMD Ryzen™ 5 2500U 2. RAM 16 GB 3. Storage 512 GB SSD + 1 TB HDD 4. VGA NVIDIA GTX 1050 Mobi le 4 GB 4.3 Pembuatan Model Perancangan pembuatan sistem untuk mencegah akses URL berbahaya menggunakan dan melatih model Large Language Model, dengan dataset yang berisi daftar situs-situs yang terindikasi berbahaya. Proses pelatihan model terdiri dari tahapan pada gambar berikut. Gambar 4.1 Tahapan Penerapan Algoritma 31 Melalui Gambar 4.1, tahap pembuatan model adalah prosedur dalam pengembangan sistem deteksi dan pencegahan akses pada URL berbahaya, menggunakan model Large Language Model dari dataset berisi daftar

situs-situs yang terindikasi berbahaya. Prosedur tahap pembuatan model meliputi pengambilan dataset, pelatihan model dan pengujian model. Berikut penjelasan terkait langkah-langkah pada gambar diatas. 4.3.1 Dataset Sumber dataset yang diperlukan berasal dari situs web kaggle.com, yang memiliki parameter data sesuai dengan kebutuhan dalam penelitian ini. Dataset ini bernama “malicious_phish.csv , parameter dataset terdiri dari; URL dan Type dengan total sebanyak 651.207 data. Parameter URL mengacu pada penulisan dari setiap situs web pada umumnya di internet. Selanjutnya, parameter Type mengacu pada sifat situs web yang terdaftar, yakni; Benign (Aman, dengan jumlah sebanyak 428.103 data), Defacement (Berbahaya, dengan jumlah sebanyak 96.457 data), Phishing (Berbahaya, dengan jumlah sebanyak 94.111 data) dan Malware (Berbahaya, dengan jumlah sebanyak 32.520 data). Kemudian, dataset ini digunakan untuk melatih model Large Language Model untuk mengidentifikasi pola penulisan URL, mengelompokkan URL yang bersifat aman dan berbahaya, serta melakukan prediksi adanya potensi URL yang berbahaya namun belum teridentifikasi. 4.3.2 Pre-Processing Data (Pra-Pemrosesan Data) Perancangan model memerlukan kebutuhan data yang sesuai untuk memastikan sistem berjalan dengan baik. Langkah-langkah pre-processing data terdiri dari; pengumpulan data, pembersihan data, pemberian label pada data, normalisasi data dan penyimpanan data yang sudah diproses. Pre-processing dilakukan untuk menyiapkan dataset yang berisi situs web aman dan tidak aman, serta memastikan data yang diproses sesuai dengan kriteria yang dibutuhkan. Berikut ini adalah gambar alur dalam pre-processing data. 32 Merujuk pada gambar 4.2, tahap-tahap pre-processing dapat dijelaskan sebagai berikut; a) Data Collection Langkah pertama pada pre-processing adalah mengumpulkan data mentah dari berbagai sumber, yakni pada penelitian ini berasal dari situs web kaggle.com. Data diambil dengan format file CSV, yang kemudian menjadi berkas untuk melatih model. b) Data Cleaning Tahap data cleaning adalah proses pembersihan data mentah dari

elemen- elemen yang tidak relevan, seperti data duplikasi, nilai kosong dan kesalahan format yang dapat mengganggu kinerja model. Gambar 4.2 Pre-processing 33 c) Data Labeling Dalam model supervised learning, yakni dalam penelitian ini berupa Large Language Model, setiap data perlu diberikan label menurut target klasifikasinya, untuk menentukan URL yang diklasifikasikan sebagai “aman” (0) atau berbahay a (1). d) Text Normalization Tujuan text normalization adalah membuat format data teks menjadi seragam, dalam hal ini; mengubah semua huruf menjadi huruf kecil, menghapus karakter khusus dan melakukan tokenisasi agar dapat diolah dengan model berbasis teks. e) Feature Extraction Data yang sudah ditokenisasi perlu dijadikan representasi numerik, guna melanjutkan proses oleh algoritma machine learning. Pada penelitian ini, tokenizer berbasis model LLM menggunakan BERT Tokenizer. f) Data Balancing Tahap ini adalah proses penyesuaian distribusi jumlah data antar kelas agar model yang dilatih tidak mengalami bias. Teknik yang digunakan pada tahap ini yaitu; - Oversampling: Jika kelas data URL “aman” lebih sedikit dari kela s data URL “berbahaya”, maka teknik ini digunakan untuk menambahkan data yang memiliki jumlah lebih sedikit (yakni URL “aman”) untu k menyeimbangkan jumlah data yang lebih banyak (yakni URL “berbahaya ”). - Undersampling: Teknik ini berfungsi untuk mengurangi jumlah data yang lebih banyak (yakni URL “berbahaya”) dengan URL “aman ”, guna menyeimbangkan jumlah data yang selanjutnya akan dilatih oleh model. 34 - Class Weighting: Berguna untuk memberikan bobot yang lebih besar pada jumlah data yang lebih sedikit, agar model lebih memperhatikan kesalahan pada jumlah data yang lebih sedikit. g) Padding/Truncation Padding/Truncation digunakan untuk menyamakan /menyeragamkan panjang input teks agar dapat diproses oleh BERT. Transformer memerlukan input dengan panjang yang seragam, maka URL yang memiliki panjang tulisan yang kurang dari batas maksimal akan diberikan padding, atau penambahan token khusus. Batas maksimal token

adalah sebanyak 64, jika URL pada data hanya berisi “www.contohwebsite.com (jumlah token tidak mencapai 64), maka URL tersebut akan dibuat menjadi “[CLS] www . contohwebsite . com [SEP] [PAD] [PAD] [PAD]….. hingga mencapai 64 token. Sedangkan jika URL pada data berisi www.contohwebsite.com/login/ 1/2/3/4/5/6/7/8/9/0..... (jumlah token melebihi 64), maka URL tersebut akan dipotong hingga panjang maksimal yang ditentukan menggunakan Truncation (pemotongan). h) Data Splitting Data splitting berperan untuk membagi dataset menjadi tiga bagian yakni; pelatihan data (training). validasi data (validation, dan pengujian data (testing). Langkah ini diperlukan untuk memastikan model dilatih secara bertahap dan dapat dievaluasi. i) Saved Pre-processed Data Setelah pre-processing selesai, data disimpan dalam bentuk format tertentu agar dapat digunakan kembali untuk pelatihan model tanpa harus pre- processing dari awal kembali. 4.3.3 Normalisasi Data Normalisasi data digunakan untuk melakukan normalisasi atau diseragamkan, pemerataan panjang URL dengan padding dan truncation untuk 35 membatasi panjang URL agar dapat disesuaikan dengan input yang diperlukan oleh BERT. 4.3.4 Split Data Langkah split data digunakan untuk memisahkan data untuk dibagi ke dalam bentuk subset, guna untuk melatih model. Subset data ini terbagi menjadi 2 bagian yakni, data latih (training set) dan data uji (testing set) dalam bentuk proporsi 80:20. 80% data digunakan model untuk mempelajari pola, hubungan dan karakteristik pada data, guna memahami konteks data dengan baik. 20% data disisihkan selama pelatihan model, guna untuk menguji kinerja model secara objektif. Hasil dari pengujian kinerja model ini akan membantu dalam generalisasi terhadap data baru yang akan diberikan selanjutnya untuk pelatihan. 4.3.5 Training Data Training data adalah langkah dalam pelatihan untuk model mengidentifikasi label “aman” dan “berbahaya” dalam dataset URL. Dis ini model mempelajari pola, struktur dan ciri-ciri pada URL yang teridentifikasi aman dan berbahaya. Semakin baik data yang diolah/

diberikan, efektifitas model dalam mengenali dan mengklasifikasikan URL semakin baik. 4.3.6 Training Model LLM Model Large Language Model (LLM) berupa BERT dilatih dalam metode training data. Proses pelatihan ini dimulai dari perubahan URL menjadi bentuk representasi numerik menggunakan tokenizer, dan diproses oleh modl untuk mempelajari konteks kata atau karakter dalam URL. Proses ini bertujuan agar model dapat mengenal dan membedakan karakteristik URL berbahaya dan URL aman secara otomatis, dengan berdasarkan pola yang dipelajari dalam proses pelatihan. 36 4.3.7 Model LLM Setelah selesai dilatih, model LLM yang sudah belajar dari data pelatihan disimpan sebagai model yang siap dipakai. Model yang telah dilatih sudah dapat melakukan prediksi dan menentukan URL yang aman atau berbahaya, serta akan diintegrasikan ke dalam SIEM sebagai sistem deteksi. 4.3.8 Testing Data Testing data adalah proses untuk melakukan evaluasi performa model, dengan cara mengambil bagian dari dataset yang tidak digunakan selama proses pelatihan. Hasil dari evaluasi dengan testing data ini dapat memberikan gambaran objektif terhadap seberapa baik model dalam menggeneralisasi dan menerapkan prediksi terhadap data yang baru. Testing data juga diberikan label sehingga hasil prediksi dapat dibandingkan dengan hasil sebenarnya. 4.3.9 Testing Model LLM Model LLM telah dilatih dan diuji menggunakan data pengujian (testing data). Tahap ini digunakan untuk menguji kinerja model dalam mendeteksi URL berbahaya berdasarkan data yang belum diketahui. 5 Metrik evaluasi yang digunakan adalah akurasi, precision, recall dan F1-score untuk memastikan model dapat mengklasifikan URL dengan baik dan efektif. 4.3.10 Sistem Deteksi dan Pencegahan URL berbahaya dengan LLM Sesudah melakukan uji kinerja dan evaluasi pada model, langkah selanjutnya adalah mengintegrasikan model LLM yang sudah dilatih ke dalam sistem SIEM melalui API Flask, bersamaan dengan Squid Proxy sebagai sistem pemblokiran jika terdeteksi adanya akses pada URL berbahaya. Berikut ini adalah alur

pemblokiran jika terjadi adanya akses pada URL berbahaya. 37 Merujuk pada Gambar 4.3, berikut adalah penjelasan proses blokir URL yang dilakukan. Pertama, client mengakses URL menggunakan aplikasi web browser, kemudian URL yang diakses oleh client akan dicegat/ intercept oleh Squid Proxy. Squid proxy melanjutkan permintaan URL oleh client kepada Flask API, yang bertugas memanggil model LLM (BERT) untuk mengklasifikasikan URL yang diakses oleh client. Jika hasil klasifikasi dari model LLM (BERT) dinyatakan aman, maka URL yang diminta mendapatkan izin akses. Jika hasil klasifikasi dari model LLM (BERT) dinyatakan berbahaya/malicious, maka izin akses URL tersebut ditolak. Kemudian, semua aktivitas tersebut dicatat pada log dalam Wazuh untuk dianalisa. Berikut ini adalah gambar pemicu payload untuk memeriksa sifat URL yang diakses. Gambar 4.3 Proses Pemblokiran URL 38 Gambar 4.4 Payload Pemicu Berikut adalah penjelasan pada Gambar 4.4; - curl -X POST: Mengirim HTTP request dengan metode POST. - http://192.168.1.100:5000/predict: Alamat IP dan port Flask API + endpoint /predict untuk klasifikasi URL. - -H "Content-Type: application/json : Header HTTP yang menunjukkan bahwa data yang dikirim berbentuk JSON. - -d '{"URL":"br-icloud.com.br"}': Data JSON yang dikirim ke model, yaitu URL br-icloud.com.br (dicurigai sebagai URL phishing/malicious). - prediction: "Malicious" adalah hasil dari Model BERT mengklasifikasikan URL tersebut sebagai URL berbahaya. - URL: "br-icloud.com.br" merupakan URL yang diuji/diberikan sebagai input. 4.4 Perancangan Sistem Pengembangan sistem pada penelitian ini memerlukan rancangan yang terstruktur untuk mengembangkan sistem deteksi dan pencegahan akses pada URL berbahaya. Rancangan ini akan direpresentasikan dalam bentuk; flowchart, activity gambar dan use case gambar. 39 4.4.1 Flowchart Proses Sistem Deteksi dan Blokir Merujuk pada Gambar 4.5 Flowchart Sistem Deteksi dan Blokir, rangkaian proses menunjukkan cara kerja sistem dalam melakukan deteksi dan blokir pada URL berbahaya. Langkah awal yang dilakukan adalah

ketika pengguna mengakses suatu URL, maka permintaan akses URL diterima oleh Squid Proxy, dilanjutkan kepada Flask API yang memanggil model LLM untuk melakukan klasifikasi terhadap URL yang diakses. Jika LLM tidak mendeteksi adanya URL berbahaya, maka akses ke internet diizinkan. Jika model LLM mendeteksi adanya URL berbahaya yang diakses, maka Squid Proxy memblokir akses pada URL berbahaya, log blokir diteruskan ke SIEM Wazuh dan menampilkan peringatan/alert. Kemudian, semua log akses terhadap URL aman dan berbahaya, tetap dapat dipantau pada SIEM Wazuh. Gambar 4.5 Flowchart Sistem Deteksi dan Blokir 40 4.4.2 Use Case Gambar Gambar 4.6 memvisualisasikan cara kerja sistem dengan use case gambar dalam sistem deteksi dan blokir akses pada URL berbahaya. Terdapat 3 actor dalam sistem ini yakni; User, Administrator dan System. User adalah actor/pengguna yang mengakses URL pada aplikasi penjelajahan situs web. Administrator adalah actor yang bertugas memantau dashboard SIEM, melihat log pada jaringan melalui SIEM dan menerima peringatan/alert jika System (SIEM) mendeteksi adanya akses pada URL berbahaya. Berikut adalah tabel skenario penggunaan use case. Tabel 4.3 Accessing URL 1. Use Case: Accessing URL 2. Deskripsi: User melakukan akses internet 3. Aktor: User 4. Action: 1. User membuka Web Browser. 2. User mengakses URL secara acak/ sesuai keinginan. Gambar 4.6 Use Case 41 Tabel 4.4 Dashboard Monitoring 1. Use Case: Dashboard Monitoring 2. Deskripsi: Administrator melakukan monitoring pada SIEM 3. Aktor: Administrator 4. Action: 1. Administrator membuka SIEM melalui Web Browser. 2. Administrator memantau Dashboard Monitoring pada SIEM. Tabel 4.5 View Network Log 1. Use Case: View Network Log 2. Deskripsi: Administrator melihat Network Log pada SIEM 3. Aktor: Administrator 4. Action: 1. Administrator membuka SIEM melalui Web Browser. 2. Administrator melihat Network Log pada SIEM. Tabel 4.6 Receiving Alert 1. Use Case: Receiving Alert 2. Deskripsi: Administrator

menerima peringatan/alert pada SIEM 3. Aktor: Administrator 4. Action: 1. Administrator membuka SIEM melalui Web Browser. 2. Administrator menerima peringatan/alert yang dikirim oleh SIEM jika terdeteksi adanya akses pada URL berbahaya. Tabel 4.7 Intrusion Detection 1. Use Case: Intrusion Detection 2. Deskripsi: System (SIEM) memantau aktivitas pada lalu lintas jaringan 3. Aktor: System (SIEM) 4. Action: System (SIEM) melakukan deteksi intrusi dan menjalankan model LLM berdasarkan file ossec.conf dan local_rules.xml yang telah dikonfigurasi 42 Tabel 4.8 Classifying URL using trained LLM 1. Use Case: Classifying URL using trained LLM 2. Deskripsi: System (SIEM) mengklasifikasikan URL yang diakses oleh User 3. Aktor: System (SIEM) 4. Action: 1. System (SIEM) menjalankan service “llm _malicious_block dengan Flask API yang sudah dikonfigurasi pada file ossec.conf. 2. Model LLM mengklasifikasikan URL yang diteruskan oleh F;ask. Jika URL terdeteksi aman, maka hasil deteksi “malicious”, jika URL terdeteksi berbahaya, maka hasil deteksi “benign”. Tabe l 4.9 Showing Alert & block Malicious URL access 1. Use Case: Showing Alert & block Malicious URL access 2. Deskripsi: System (SIEM) menunjukkan peringatan akses dan blokir pada URL berbahaya 3. Aktor: System (SIEM) 4. Action: 1. Squid Proxy mendeteksi adanya akses terhadap URL Berbahaya melalui verifikasi dari Flask API yang menjalankan “llm_malicious_block dengan model LLM 2. Akses pada URL berbahaya yang diakses oleh User diblokir dan menunjukkan halaman tidak tersedia/terblokir oleh Flask API 43 4.4.3 Activity Gambar Activity gambar pada penelitian ini berfungsi untuk menjabarkan urutan aktivitas dalam use case yang sebelumnya telah dibuat. Berikut adalah semua activity gambar beserta penjelasannya. Merujuk pada Gambar 4.7, ada tiga peran dalam sistem deteksi dan pencegahan terhadap akses URL berbahaya yaitu; User, Sistem dan Model. Langkah pertama dimulai dari ketika User mengakses URL apapun dalam aplikasi web browser, lalu permintaan akses URL oleh User diteruskan ke Squid

Proxy sebagai pencegat/interceptor. Kemudian, Squid Proxy melanjutkan permintaan Gambar 4.7 Activity Diagram: User 44 URL kepada Flask API untuk memanggil Model LLM yang berperan dalam mengklasifikasikan sifat URL. Ketika Model LLM sudah menentukan sifat dari URL yang diminta, maka jika terdeteksi berbahaya Squid Proxy memblokir akses, mengirimkan pesan blokir kepada SIEM, memberikan peringatan/alert terkait URL yang diminta dan menunjukkan halaman Error pada User. Jika URL terdeteksi aman, maka Squid Proxy memperbolehkan akses, mengirimkan pesan perizinan akses kepada SIEM dan mengizinkan User untuk mengakses URL yang terdeteksi aman. Gambar 4.8 menunjukkan aktivitas administrator dalam mengakses sistem SIEM. Pertama, administrator menggunakan web browser dengan memanggil alamat IP milik SIEM, kemudian masuk pada sistem dan SIEM menunjukkan dashboard monitoringnya. Gambar 4.8 Activity Diagram: Dashboard Monitoring 45 Merujuk pada Gambar 4.9, proses diatas menunjukkan proses administrator mengakses menu “Network Log” pada SIEM untuk memantau aktivitas yan g terjadi pada jaringan. Gambar 4.9 Activity Diagram: View Network Log 46 Gambar 4.10 menjelaskan alur proses ketika administrator menerima alert/peringatan jika terdeteksi adanya akses pada URL berbahaya dengan cara melihat pada menu Network Log. Gambar 4.10 Activity Diagram: Receiving Alert Gambar 4.11 Activity Diagram: Intrusion Detection 47 Pada Gambar 4.11, administrator mengonfigurasi file ossec.conf dan local_rules.xml pada sistem SIEM yang berfungsi dalam pemantauan deteksi intrusi. File ossec.conf dan local_rules.xml ini juga ditambahkan dengan konfigurasi untuk SIEM dapat mendeteksi dan memblokir jika terjadi akses pada URL berbahaya. Merujuk pada Gambar 4.12, permintaan akses URL yang dilakukan oleh user akan diproses melalui Flask API yang selanjutnya diteruskan kepada model LLM. Kemudian, model LLM akan memproses URL dan menunjukkan hasil berupa “benign”(aman) atau “malicious”(berbahaya). Gambar 4.12 Activit y Diagram: Classifying URL using trained LLM 48 Gambar 4.13

menunjukkan alur sistem menunjukkan peringatan/alert dan pemblokiran akses pada URL berbahaya. Squid Proxy mendeteksi adanya akses URL berbahaya, kemudian proses pemblokiran dimulai dari menunjukkan alert pada SIEM dan berakhir dengan menunjukkan halaman “akses diblokir ” oleh Flask API. 4.5 Skenario Pengujian Model LLM yan g suda h dilatih dan diintegrasikan ke dalam SIEM untuk mendeteksi dan mencegah akses pada URL berbahaya memerlukan verifikasi pengujian. Verifikasi dan pengujian ini merujuk pada bagian Bab III “Metode Pengujian”. Berikut adalah detail dari setiap metode pengujian yang dilakukan. Gambar 4.13 Activity Gambar: Showing Alert & block Malicious URL access 49 4.5.1 Pengujian White Box Metode pengujian white box merujuk pada algoritma pada model LLM yang sudah dilatih untuk mengklasifikasikan antara URL yang berbahaya dan yang aman. Berikut ini adalah tabel penjelasan dalam metode white box yang dilakukan. Tabel 4.10 Pengujian White Box No Modul yang Diuji Output yang Diharapkan 1 Tokenisasi URL Format input_ids dan attention_mask valid dan sesuai 2 Klasifikasi oleh Model BERT Output label malicious atau benign + confidence score 3 Flask A PI dan Squid Proxy Filtering Flask API menerima respon dan Squid Proxy memblokir akses URL berbahaya 4 Logging ke Wazuh Alert muncul pada log alerts Wazuh CLI 5 Rule Matching dan Active Response Penyesuaian Rules yang sudah ada dan penambahan rules untuk pemblokiran URL berbahaya dari Flask API dan Squid Proxy 4.5.2 Pengujian Black Box Metode pengujian black box merujuk pada performa akan ditunjukkan ketika mengeksekusi sistem yang telah dikembangkan. Berikut ini adalah tabel penjelasan dalam metode black box yang dilakukan. Tabel 4.11 Pengujian Black Box No Skenario Pengujian Hasil yang Diharapkan 1 Client mengakses URL aman Halaman ditampilkan dengan normal 2 Client mengakses URL berbahaya Akses diblokir, halaman blokir Squid tampil 50 3 Flask API menerima dan merespons klasifikasi URL JSON response dengan label benar dan waktu

respon < 1 detik 4 Log dikirim ke Wazuh Terdapat log baru di Kibana/Wazuh Alerts 5 Wazuh menampilkan alert Alert muncul sesuai rule 100101 pada ossec.conf 6 Admin memantau aktivitas Dapat melihat riwayat event akses URL, pemblokiran, dan alert 51 BAB V HASIL DAN PEMBAHASAN Bab ini menyajikan hasil sistem yang sudah dikembangkan dan dilakukan oleh peneliti, beserta dengan pembahasannya seperti berikut. 5.1 Hasil Pengembangan SIEM berbasis Wazuh untuk mencegah akses pada URL berbahaya berhasil dikembangkan menggunakan dataset yang berisi URL-URL berbahaya dalam sebuah file yang berformat CSV. Dataset tersebut berisi parameter URL dan Type yang kemudian dilatih oleh model LLM untuk melakukan prediksi serta pencegahan jika URL terdeteksi berbahaya/malicious. Hasil yang dicapai setelah proses pelatihan model membuktikan bahwa URL yang bersifat berbahaya/malicious dapat dideteksi pada SIEM dan diblokir aksesnya. Penelitian ini memanfaatkan Large Language Model berbasis BERT untuk memprediksi dan memblokir akses pada URL berbahaya. Umumnya, pemantauan dilakukan secara manual sehingga pemeriksaan akses pada URL juga belum bisa secara otomatis. Sehingga pengembangan pada penelitian ini bertujuan untuk memastikan pemblokiran pada URL berbahaya dilakukan secara otomatis. Berikut adalah hasil pelatihan dan pengujian yang telah dilakukan. Berdasarkan gambar diatas, hasil pengujian adalah sebagai berikut; Gambar 5.1 Hasil pengujian Akurasi dan Confusion Matrix pada model LLM 52 Tabel 5.1 Tabel hasil pengujian Confusion Matrix Confusion Matrix Predicted Benign Predicted Malicious True Benign 1287 16 True Malicious 11 686 Hasil pengujian menunjukkan: - True Positive/True Malicious: 686 URL berbahaya berhasil dideteksi sebagai berbahaya. - True Negative/True Benign: 1287 URL aman dan dideteksi sebagai aman. - False Positive/ Predicted Malicious: 16 URL aman tapi salah diklasifikasikan sebagai berbahaya. - False Negative/Predicted Benign: 11 URL berbahaya tapi lolos sebagai aman. Tabel 5.2 Tabel hasil pengujian Evaluation

Metrics Evaluation Metrics Accuracy 0.9865 / 98% Precision 0.9772 / 97% Recall 0.9842 / 98% F1-Score 0.9807 / 98% Hasil penguj ian menunjukkan hasil accuracy mencapai 98%; dari seluruh URL yang diuji, hasil klasifikasi antara URL berbahaya dan aman mencapai 98%. Hasil precision mencapai 97%; dari seluruh URL berbahaya yang diuji, 97% menunjukkan bahwa URL benar-benar berbahaya. Hasil recall mencapai 98%; 98% menunjukkan model berhasil memprediksi dan mendeteksi seluruh URL yang benar-benar berbahaya. Hasil F1-Score mencapai 98%; model berhasil menyeimbangkan antara metrik True Positive, True Negative, False Positive dan False Negative sebesar 98%. 53 Gambar diatas adalah file predict.py yang berfungsi untuk melakukan prediksi dan menentukan sifat URL dari dataset yang telah dilatih. Jika user/ pengguna terdeteksi mengakses URL berbahaya, maka akses akan dialihkan pada halaman blokir. Jika user/pengguna mengakses URL aman, maka akses akan dilanjutkan tanpa adanya halaman blokir Gambar 5.2 Konfigurasi predict.py 54 File llm_predict_service.service pada gambar diatas digunakan untuk mengaktivasi layanan/service yang menuju pada predict.py, sehingga fungsi untuk melakukan prediksi atau memeriksa sifat URL yang telah dilatih dapat dilakukan secara otomatis dengan perintah “curl”. Gambar 5.4 merupakan konfigurasi Squid Proxy sebagai pemblokir jika ada terjadinya akses pada URL berbahaya. Squid Proxy juga berperan untuk melanjutkan log blokir agar dapat ditampilkan pada Wazuh, dan memanggil konfigurasi URL_checker.sh yang berfungsi untuk memeriksa hasil pemeriksaan URL melalui konfigurasi predict.py. Gambar 5.3 Konfigurasi llm_predict_service.service Gambar 5.4 Konfigurasi Squid Proxy 55 Konfigurasi URL_checker.sh berfungsi sebagai penghubung/API antara predict.py dan squid.conf. Jika ada permintaan akses pada URL dan dilanjutkan kepada predict.py, maka URL_checker.sh melanjutkan kepada squid.conf untuk menentukan hasil prediksi URL yang telah dilakukan oleh predict.py. Jika predict.py menunjukkan hasil URL yang berbahaya, maka URL_checker.sh akan

melanjutkan hasil URL tersebut sebagai berbahaya dan diblokir oleh Squid melalui konfigurasi squid.conf. Gambar 5.5 Konfigurasi URL _checker.sh 56 Merujuk pada gambar 5.6, konfigurasi ossec.conf ditujukan agar Wazuh dapat menerima log dalam bentuk .json dan menunjukkan log pemblokiran yang dilanjutkan oleh konfigurasi URL _checker.sh. Gambar 5.6 Konfigurasi ossec.conf 57 Konfigurasi local _rules.xml digunakan untuk membuat aturan khusus dan menulis log buatan/custom pada Wazuh jika terjadi adanya akses pada URL berbahaya. Nomor identifikasi aturan ini adalah 100100, dan ketika dimunculkan pada Wazuh, maka nomor identifikasi akan ditunjukkan beserta dengan pesan pemblokiran buatan/custom yang telah dibuat. Gambar 5.7 Konfigurasi local_rules.xml 58 Halaman login Wazuh ditampilkan ketika mengakses alamat IP lokal yang dikonfigurasi secara otomatis. Kredensial bawaan yang terpasang pada Wazuh ini adalah; username: “admin” dan password “admin”. Setelah melakukan login, a kan muncul tampilan seperti pada gambar 5.9. Dalam penelitian ini, menu yang digunakan yaitu adalah “Threat Hunting”, guna untuk menunjukkan log hasil pemblokiran URL. Gambar 5.8 Halaman login Wazuh Gambar 5.9 Dashboard Wazuh 59 Gambar 5.10 adalah dashboard yang digunakan untuk memantau log, merujuk pada penelitian ini adalah log pemblokiran akses terhadap URL berbahaya. Gambar diatas menunjukkan adanya akses pada URL berbahaya dan telah ditampikan pada “rule.description” dan “rule.id” yang sesuai dengan konfigurasi l ocal_rules.xml dan ossec.conf. Gambar 5.10 Dashboard Threat Hunting 60 5.2 Uji Coba Hasil Pengujian Pengujian pada sistem dilakukan dengan metode White Box dan metode Black Box yang sebelumnya telah dilampirkan pada sub-bab 4.5. Berikut adalah isi dari pengujian yang telah dilakukan. 5.2.1 Hasil Pengujian White Box Tabel 5.3 Pengujian White Box No Modul yang Diuji Output yang Diharapkan 1 Tokenisasi URL Format input_ids dan attention_mask valid dan sesuai Hasil: Sesuai Hasil sesuai dengan menunjukkan hasil tokenisasi pada URL

yang dimasukkan untuk melakukan prediksi “benign” atau “malicious”. 2 Klasifikasi oleh Model BERT Output label malicious atau benign Hasil: Sesuai Hasil sesuai dengan dimunculkannya hasil prediksi “verdic t”:”malicious” terhadap URL yang benar-benar berbahaya. 61 3 Flask API dan Squid Proxy Flask API menerima respon dan Squid Proxy memblokir akses URL berbahaya Hasil: Sesuai Hasil sesuai dengan dimunculkannya hasil pemblokiran yang dilakukan oleh Squid Proxy dan halaman blokir oleh Flask API. 4 Logging ke Wazuh Alert muncul pada log alerts Wazuh CLI 62 Hasil: Sesuai Hasil sesuai dengan dimunculkannya nomor identifikasi aturan/rule 100100 dan URL berbahaya yang diakses. 5 Rule Matching dan Active Response Penyesuaian Rules yang sudah ada dan penambahan rules untuk pemblokiran URL berbahaya dari Flask API dan Squid Proxy Hasil: Sesuai Hasil sesuai merujuk pada konfigurasi yang ditunjukkan pada gambar dan berhasil dimunculkan pada Wazuh. 63 5.2.2 Hasil Pengujian Black Box Tabel 5.4 Pengujian Black Box No Skenario Pengujian Hasil yang Diharapkan 1 Client mengakses URL aman Halaman ditampilkan dengan normal (youtube.com) Hasil: Sesuai Hasil sesuai dengan menunjukkan halaman web sah “youtube.com” yang bisa diakses. 2 Client mengakses URL berbaha ya Akses diblokir, halaman blokir Squid tampil 64 Hasil: Sesuai Hasil sesuai dengan percobaan akses kepada URL berbahaya dan akses terblokir. 3 Flask API menerima dan merespons klasifikasi URL Flask API menerima permintaan dan mengembalikan halaman Akses Diblokir 65 Hasil: Sesuai Hasil sesuai dengan merujuk pada Flask API yang menunjukkan halaman pemblokiran. 4 Log dikirim ke Wazuh dan menampilkan alert Alert muncul sesuai rule 100100 pada local_ rules.xml 66 Hasil: Sesuai Hasil sesuai ditunjukkan dengan adanya aturan/rule khusus yang dibuat untuk pemblokiran akses pada URL berbahaya. 67 BAB VI PENUTUP Kesimpulan dan hasil yang telah diselesaikan dalam penelitian ini, dengan referensi serta teori yang telah disampaikan pada bab-bab sebelumnya akan disampaikan dalam bab ini. 9 Saran

serta rekomendasi pada penelitian ini juga akan dipaparkan untuk pengembangan kedepannya. 6.1 Kesimpulan Dengan menggabungkan teknologi Security Information and Event Management (SIEM) berbasis Wazuh, model klasifikasi URL berbasis BERT, dan Squid Proxy Server untuk mekanisme pemblokiran, penelitian ini berhasil merancang dan menerapkan sistem deteksi dan pencegahan akses terhadap URL berbahaya. Sebagai hasil dari pengujian yang dilakukan, dapat disimpulkan bahwa: 1. Dengan menggunakan dataset URL publik, model klasifikasi URL berbasis BERT berhasil membedakan URL benign dan malicious dengan akurasi tinggi sebesar 98,65%, dengan nilai precision 97,72%, recall 98,42%, dan F1-score 98,07%. Ini menunjukkan bahwa model memiliki kemampuan untuk melakukan deteksi URL berbahaya secara konsisten dan andal. 2. Flask API berfungsi dengan baik sebagai antarmuka antara model dan Squid Proxy Server. Ketika seseorang mencoba mengakses URL tertentu, API akan mengirimkan URL tersebut ke model, dan hasil klasifikasi ("malicious" atau "benign") menentukan keputusan pemblokiran. 3. Pengguna akan diarahkan ke halaman blokir khusus yang ditampilkan melalui web server Flask dengan URL berbahaya, sementara dengan URL aman, akses akan diteruskan sebagaimana mestinya. 4. Semua modul sistem, termasuk tokenisasi URL, klasifikasi, filtering proxy, pengiriman log, dan visualisasi dashboard, bekerja sesuai harapan dengan pengujian white box dan black box. 68 6.2 Saran Beberapa rekomendasi untuk pengembangan dan perbaikan sistem di masa depan meliputi: 1. Penanganan HTTPS Secara Menyeluruh: Squid Proxy saat ini tidak dapat melakukan inspeksi HTTPS secara transparan. Untuk menginspeksi konten HTTPS lebih dalam, dapat menggunakan SSL Bump atau proxy chaining dengan layer filtering lainnya. 2. Model Tambahan: Model BERT dapat disempurnakan untuk meningkatkan kinerja klasifikasi dengan menggunakan teknik fine-tuning pada kumpulan data yang lebih besar. 3. Pengujian Lebih Luas di Jaringan Produksi: Disarankan untuk menguji sistem pada skala jaringan yang lebih besar

dan kompleks (dengan banyak klien) untuk memastikan skalabilitas dan performa sistem dalam kondisi nyata. 4. Auto-Update Threat Intelligence: Implementasikan sistem untuk memperbarui model atau blacklist URL secara otomatis dari sumber threat intelligence seperti PhishTank, Google Safe Browsing, atau URLHaus.

REPORT #27589607

Results

Sources that matched your submitted document.

IDENTICAL CHANGED TEXT

1.	INTERNET SOURCE **0.45%** elibrary.unikom.ac.id *https://elibrary.unikom.ac.id/id/eprint/477/10/UNIKOM_YOGA%20EKA%20PERM..*	
2.	INTERNET SOURCE **0.41%** repository.unhas.ac.id *http://repository.unhas.ac.id/id/eprint/44254/1/H071201001_skripsi_25-09-2024..*	
3.	INTERNET SOURCE **0.3%** eprints.upj.ac.id *https://eprints.upj.ac.id/id/eprint/7526/13/BAB%20II.pdf*	
4.	INTERNET SOURCE **0.28%** eprints.ums.ac.id *https://eprints.ums.ac.id/30442/4/BAB_I.pdf*	
5.	INTERNET SOURCE **0.22%** www.academia.edu *https://www.academia.edu/98369535/Analisa_Learning_Rate_dan_Batch_Size_...*	
6.	INTERNET SOURCE **0.19%** eprints.upj.ac.id *https://eprints.upj.ac.id/id/eprint/4088/19/BAB%20III.pdf*	
7.	INTERNET SOURCE **0.17%** medium.com *https://medium.com/@audreyroselian02/ulasan-bert-pre-training-of-deep-bidir...*	
8.	INTERNET SOURCE **0.17%** www.ibm.com *https://www.ibm.com/id-id/think/topics/context-window*	
9.	INTERNET SOURCE **0.13%** repository.uinjkt.ac.id *https://repository.uinjkt.ac.id/dspace/bitstream/123456789/66813/1/DINO%20A...*	

REPORT #27589607

INTERNET SOURCE

10. **0.12%** jurnal.umitra.ac.id

https://jurnal.umitra.ac.id/index.php/JMA/article/download/1854/1581