# HexaSLM: Bisakah Chain-of-Verification Mengamankan Domain Keamanan Siber?

## Abstrak

Integrasi *Large Language Models* (LLM) ke dalam operasi keamanan siber menghadirkan paradoks yang signifikan: sementara kemampuan mereka dalam deteksi ancaman otomatis, analisis kode, dan respons insiden bersifat transformatif, kecenderungan mereka untuk mengalami halusinasi—menghasilkan informasi yang terdengar masuk akal namun secara faktual salah—menimbulkan risiko yang tidak dapat diterima dalam lingkungan keamanan berisiko tinggi. Makalah ini memperkenalkan **HexaSLM**, sebuah *Small Language Model* (SLM) berparameter 1,5 miliar yang dikhususkan untuk memitigasi halusinasi melalui integrasi novel antara *Chain-of-Verification* (CoVe) dan *instruction tuning* spesifik domain. Memanfaatkan arsitektur **Qwen 2.5 Instruct** sebagai dasar, HexaSLM dilatih menggunakan metodologi **Unsloth** dan **QLoRA** *Parameter-Efficient Fine-Tuning* (PEFT), menggunakan kurikulum campuran yang terdiri dari **10.000 sampel penalaran intensif dari dataset PRM800K** yang digabungkan dengan data tahap kedua dari **AlicanKiraz0/Cybersecurity-Dataset-v1**.

Hasil kami menunjukkan bahwa HexaSLM mencapai **akurasi state-of-the-art sebesar 82,0%** di seluruh tolok ukur keamanan siber yang ketat, mencakup OWASP Top 10, Panduan NIST, dan Kerentanan Umum. Yang paling signifikan, model ini menunjukkan **tingkat halusinasi sebesar 14,0%**, sebuah pengurangan drastis dibandingkan dengan tingkat 42,1% dari model dasar dan 25,7% dari baseline LoRA standar, menjadikannya satu-satunya model yang berhasil menembus ambang batas keamanan 15%. Lebih lanjut, model ini menunjukkan **tingkat kepatuhan 86,0%** terhadap protokol *Chain-of-Verification*, memvalidasi efektivitas metodologi pelatihan yang diusulkan. Makalah ini merinci jalur pelatihan, menyediakan studi ablasi yang ekstensif, dan menganalisis implikasi LLM yang andal untuk penasihat keamanan otomatis. Implementasi model lengkap dan artefak pelatihan tersedia di [**https://github.com/AneKazek/HexaSLM**](https://github.com/AneKazek/HexaSLM) dan [**https://huggingface.co/anekazek/hexaslm-qwen2.5-cybersec-cove**](https://huggingface.co/anekazek/hexaslm-qwen2.5-cybersec-cove).

## 1. Pendahuluan

Lanskap keamanan siber kontemporer dicirikan oleh asimetri fundamental antara pembela (*defenders*) dan penyerang (*adversaries*). Pembela harus mengamankan permukaan serangan yang terus meluas secara eksponensial—mulai dari infrastruktur *cloud native*, perangkat IoT, hingga kode aplikasi—sementara penyerang hanya perlu menemukan satu kerentanan untuk berhasil. Dalam konteks ini, potensi Kecerdasan Buatan (AI), khususnya *Large Language Models* (LLM), untuk berfungsi sebagai pengganda kekuatan (*force multipliers*) bagi Pusat Operasi Keamanan (SOC) sangatlah besar.1 Mulai dari mengotomatiskan penguraian laporan intelijen ancaman (*Cyber Threat Intelligence* - CTI) hingga menghasilkan skrip remediasi untuk kerentanan kompleks, LLM menjanjikan pengurangan beban kognitif pada analis keamanan yang sering kali kelebihan beban kerja.

Namun, penyebaran LLM tujuan umum (*general-purpose*) dalam domain kritis keamanan sangat terhambat oleh fenomena "halusinasi"—kecenderungan model ini untuk memalsukan fakta, mengutip kerentanan yang tidak ada (misalnya, *package hallucination*), atau menghasilkan kode yang tidak aman sambil mempertahankan nada kepercayaan diri yang tinggi.2 Dalam domain keamanan siber, halusinasi bukan sekadar kesalahan teknis; itu adalah kerentanan potensial. Sebuah model yang salah mengidentifikasi file jinak sebagai berbahaya (*false positive*) menyebabkan gesekan operasional, tetapi model yang menyarankan konfigurasi yang tidak aman sebagai "perbaikan" (*false negative* atau misinformasi) secara aktif mendegradasi postur keamanan organisasi.4

Evaluasi terbaru menunjukkan bahwa bahkan model canggih seperti GPT-4 dan Llama 3 dapat menghasilkan saran keamanan yang salah atau gagal membedakan antara ID CVE asli dan palsu.5 Konsekuensinya, terdapat kebutuhan mendesak akan model khusus yang memprioritaskan *verifiabilitas* dan *kesetiaan* (*faithfulness*) di atas generasi kreatif. Masalah ini diperburuk oleh kendala sumber daya; banyak organisasi keamanan memerlukan solusi yang dapat dijalankan secara lokal (on-premise) untuk menjaga privasi data, yang membatasi penggunaan model parameter raksasa berbasis cloud.

Penelitian ini menghadirkan **HexaSLM**, sebuah *Small Language Model* (SLM) yang dibangun khusus untuk mengatasi tantangan keandalan ini. Hipotesis yang mendorong pekerjaan ini adalah bahwa integrasi struktur penalaran eksplisit—secara spesifik **Chain-of-Verification (CoVe)** 7—ke dalam proses pelatihan model yang lebih kecil dan lebih efisien dapat menghasilkan kinerja yang kompetitif dengan atau bahkan lebih unggul dari model generalis yang lebih besar dalam domain tertentu. Dengan melakukan *fine-tuning* pada model **Qwen 2.5 1.5B** 9 menggunakan **Low-Rank Adaptation (LoRA)** 10 dan pustaka optimasi **Unsloth** 12, penelitian ini mencapai keseimbangan antara efisiensi komputasi dan penalaran fidelitas tinggi.

Kontribusi utama dari makalah ini adalah sebagai berikut:

1. **Inovasi Metodologis:** Kami mengusulkan kurikulum pelatihan hibrida yang mencampurkan **10.000 sampel data penalaran yang diawasi proses (Process Supervision)** dari dataset PRM800K 14 dengan instruksi keamanan siber spesifik domain.16 Pendekatan ini menanamkan pengetahuan domain model ke dalam struktur verifikasi yang ketat.
2. **Validasi Empiris:** Kami menyediakan evaluasi komprehensif di empat kategori keamanan kritis—Praktik Terbaik (*Best Practices*), Kerentanan Umum (*Common Vulnerabilities*), Panduan NIST, dan OWASP Top 10—yang menunjukkan akurasi unggul (82,0%) dan pengurangan halusinasi yang signifikan (14,0%) dibandingkan dengan baseline yang kuat.17
3. **Reproducibility Terbuka:** Sejalan dengan prinsip sains terbuka, kami merilis kode lengkap, dataset, dan bobot model melalui GitHub dan HuggingFace, memfasilitasi penelitian lebih lanjut tentang AI keamanan yang tepercaya.

## 2. Tinjauan Pustaka

### 2.1 Model Bahasa Besar dalam Keamanan Siber

Penerapan Pemrosesan Bahasa Alami (NLP) pada keamanan siber telah berkembang dari algoritma pencocokan pola sederhana dan berbasis tanda tangan (*signature-based*) menuju model berbasis transformer yang canggih. Karya-karya awal seperti **SecureBERT** 18 dan **CyBERT** mendemonstrasikan kegunaan pra-pelatihan adaptif domain untuk tugas-tugas seperti klasifikasi Intelijen Ancaman Siber (CTI) dan penilaian kerentanan. Namun, arsitektur *encoder-only* ini tidak memiliki kemampuan generatif yang diperlukan untuk tugas penalaran dan sintesis yang kompleks, seperti menjelaskan rantai serangan atau menyarankan perbaikan kode.

Pergeseran menuju arsitektur *decoder-only* (misalnya, seri GPT, Llama, Qwen) membuka kemungkinan baru untuk respons insiden otomatis dan pembuatan kode. Survei terbaru 1 menyoroti bahwa meskipun LLM unggul dalam deteksi ancaman dan analisis log, mereka menderita keterbatasan signifikan mengenai interpretabilitas dan konsistensi faktual. Secara khusus, model sering gagal mematuhi standar industri yang kaku seperti **NIST Risk Management Framework** 21 atau panduan **OWASP** 23, yang mengarah pada keluaran yang "masuk akal secara keamanan" tetapi tidak patuh (*non-compliant*). Penelitian ini mengatasi kesenjangan ini dengan secara langsung menanamkan kerangka kerja ini ke dalam logika verifikasi model.

### 2.2 Mitigasi Halusinasi dan Chain-of-Verification

Halusinasi dalam LLM adalah mode kegagalan yang terdokumentasi dengan baik, sangat berbahaya dalam domain tertutup (*closed domains*) seperti hukum, kedokteran, dan keamanan.2 Strategi mitigasi tradisional berfokus pada *Retrieval-Augmented Generation* (RAG) 26 untuk mendasarkan generasi pada pengetahuan eksternal. Meskipun efektif, sistem RAG sangat kompleks untuk dipelihara, memerlukan infrastruktur basis data vektor, dan masih dapat menderita kesalahan pengambilan (*retrieval errors*) atau keracunan konteks.

**Chain-of-Verification (CoVe)**, yang diperkenalkan oleh Dhuliawala dkk. (2023) 7, mengusulkan strategi mitigasi intrinsik. CoVe memaksa model untuk menyusun draf respons awal, merencanakan pertanyaan verifikasi untuk memeriksa pekerjaannya sendiri, menjawab pertanyaan-pertanyaan tersebut secara independen, dan kemudian menghasilkan respons akhir yang telah diverifikasi. Proses berpikir "Sistem 2" ini telah terbukti mengurangi halusinasi secara signifikan dalam tugas domain umum. Pekerjaan kami adalah salah satu yang pertama menerapkan dan menyempurnakan CoVe secara sistematis khusus untuk domain keamanan siber, memvalidasi keefektifannya dalam mengurangi misinformasi keamanan yang berbahaya.

### 2.3 Pengawasan Proses (***Process Supervision***) dan Dataset PRM800K

Konsep **Process Supervision**, yang diperjuangkan oleh Lightman dkk. (2023) 15, berpendapat bahwa memberi penghargaan pada model untuk *langkah penalaran* yang benar lebih efektif daripada memberi penghargaan hanya untuk hasil akhir (*Outcome Supervision*). Dataset **PRM800K** 14 menyediakan label tingkat langkah untuk tugas penalaran kompleks. HexaSLM memanfaatkan subset dari dataset ini untuk mengajarkan model *struktur* verifikasi langkah demi langkah yang ketat. Hipotesis kami adalah bahwa pola penalaran yang diperlukan untuk memecahkan masalah matematika mentransfer secara efektif ke deduksi logis yang diperlukan untuk analisis keamanan (misalnya, menelusuri aliran data *Reflected XSS* atau memvalidasi logika otentikasi).

### 2.4 Fine-Tuning Efisien dengan QLoRA dan Unsloth

Melatih LLM berkinerja tinggi secara tradisional membutuhkan sumber daya komputasi yang masif, seringkali di luar jangkauan laboratorium penelitian akademis atau tim keamanan perusahaan kecil. **LoRA (Low-Rank Adaptation)** 11 merevolusi paradigma ini dengan membekukan bobot pra-pelatihan dan menyuntikkan matriks dekomposisi peringkat yang dapat dilatih. **QLoRA** 10 lebih lanjut mengoptimalkan ini dengan memperkenalkan kuantisasi *4-bit NormalFloat* (NF4), memungkinkan *fine-tuning* model miliaran parameter pada perangkat keras tingkat konsumen tanpa degradasi kinerja yang signifikan.

Pustaka **Unsloth** 12 membawa efisiensi ini ke tingkat berikutnya dengan kernel Triton yang ditulis tangan, mencapai kecepatan pelatihan hingga 2x lebih cepat dan pengurangan memori 70%. Unsloth mengoptimalkan operasi *backpropagation* secara manual dan menangani perhitungan gradien dengan presisi tinggi meskipun model dimuat dalam 4-bit. HexaSLM menggunakan tumpukan teknologi ini untuk mendemonstrasikan bahwa model keamanan berkualitas tinggi dapat dilatih secara aksesibel, mendemokratisasi kemampuan pertahanan siber tingkat lanjut.32

## 3. Kerangka Teoretis

Hipotesis sentral dari penelitian ini adalah bahwa keahlian keamanan siber dapat dimodelkan sebagai sistem proses ganda: **pengambilan pengetahuan** (mengetahui apa itu injeksi SQL) dan **verifikasi aplikasi** (mengonfirmasi bahwa cuplikan kode tertentu rentan dan perbaikannya aman). *Instruction tuning* standar meningkatkan yang pertama tetapi sering mengabaikan yang kedua.

### 3.1 Formalisme Chain-of-Verification

Secara formal, misalkan  adalah model bahasa dan  adalah pertanyaan pengguna. Dalam generasi standar, , di mana  adalah respons. Dalam CoVe, proses didekomposisi menjadi empat langkah 7:

1. **Drafting (Penyusunan):** . Model menghasilkan jawaban awal yang mungkin mengandung halusinasi.
2. **Planning (Perencanaan):** , di mana  adalah pertanyaan verifikasi yang menargetkan klaim faktual dalam .
3. **Execution (Eksekusi):** , di mana  adalah jawaban untuk pertanyaan verifikasi . Langkah ini idealnya dilakukan secara independen untuk menghindari bias konfirmasi.
4. **Synthesis (Sintesis):** . Model merevisi draf awal berdasarkan hasil verifikasi untuk menghasilkan output akhir yang akurat.

HexaSLM disetel (fine-tuned) untuk menginternalisasi langkah 2 dan 3 secara eksplisit dalam aliran generasi outputnya. Dengan melatih pada subset **PRM800K**, model belajar menghargai *kebenaran langkah perantara*. Dengan melatih pada **AlicanKiraz0/Cybersecurity-Dataset-v1**, model mempelajari *semantik domain* ( dan ). Fusi dataset ini menciptakan model yang memperlakukan nasihat keamanan bukan sebagai tugas penulisan kreatif, tetapi sebagai derivasi kebenaran langkah demi langkah.

## 4. Metodologi

### 4.1 Pemilihan Model Dasar: Qwen 2.5

Kami memilih **Qwen 2.5 1.5B Instruct** 9 sebagai fondasi untuk HexaSLM. Seri Qwen 2.5 terkenal karena kinerjanya yang luar biasa pada tolok ukur pengkodean dan matematika relatif terhadap ukurannya, menjadikannya kandidat ideal untuk domain keamanan siber yang sarat logika. Ukuran parameter 1,5 miliar dipilih secara strategis untuk mendemonstrasikan kelayakan penyebaran agen keamanan yang sangat mampu pada perangkat *edge* atau dalam lingkungan SOC yang terbatas sumber daya, meminimalkan latensi inferensi dan biaya operasional.34 Model ini mendukung *context window* hingga 32k token, yang penting untuk analisis log atau kode, meskipun kami membatasi pelatihan pada 2048 token untuk efisiensi.17

### 4.2 Kurasi dan Sintesis Dataset

Data pelatihan untuk HexaSLM dibangun melalui strategi pencampuran strategis yang dirancang untuk menyeimbangkan kemampuan penalaran abstrak dengan pengetahuan domain konkret.

#### 4.2.1 Tahap 1: Injeksi Penalaran (PRM800K)

Untuk menanamkan pola pikir verifikasi yang ketat, kami memanfaatkan subset dari **10.000 sampel** dari dataset **PRM800K**.14 Meskipun PRM800K utamanya adalah dataset matematika (berasal dari dataset MATH), strukturnya—yang berisi solusi langkah demi langkah dengan label kebenaran granular—adalah isomorfik dengan penelusuran logis yang diperlukan dalam keamanan siber (misalnya, analisis jalur serangan atau *kill chain*). Dengan melatih model untuk menghormati logika bertahap (step-wise logic), kami mempersiapkannya untuk mekanisme CoVe. Penggunaan 10.000 sampel dipilih sebagai keseimbangan optimal untuk mentransfer gaya penalaran tanpa membuat model melupakan kemampuan bahasanya atau menjadi terlalu bias terhadap notasi matematika.

#### 4.2.2 Tahap 2: Adaptasi Domain (AlicanKiraz0/Cybersecurity-Dataset-v1)

Spesifisitas domain dicapai menggunakan **AlicanKiraz0/Cybersecurity-Dataset-v1**.16 Dataset ini berisi ribuan pasangan instruksi-respons berkualitas tinggi yang berfokus pada pendidikan keamanan siber defensif. Secara krusial, dataset ini mencakup pola "penolakan" (*refusal patterns*) untuk permintaan berbahaya (misalnya, pembuatan *ransomware* atau *phishing kits*), menyelaraskan model dengan **Kode Etik ISC** dan prinsip utilitas pertahanan saja (*defense-only*). Ini memastikan bahwa HexaSLM bertindak sebagai asisten "Blue Team" yang bertanggung jawab daripada senjata "Red Team".

Logika pencampuran melibatkan *interleaving* dataset ini untuk mencegah *catastrophic forgetting*. Model dipaksa untuk menerapkan pola penalaran dari PRM800K ke konten dataset keamanan siber, secara efektif "menulis ulang" pengetahuan keamanan ke dalam format yang diverifikasi.

### 4.3 Pipa Pelatihan (***Training Pipeline***)

Model disetel menggunakan pustaka **Unsloth** 12 pada lingkungan Linux yang dilengkapi dengan **2x GPU NVIDIA Tesla T4** (masing-masing 14,6 GB VRAM). Konfigurasi pelatihan adalah sebagai berikut:

* **Kerangka Kerja:** Unsloth 2026.2.1, Transformers 4.57.1, PyTorch 2.8.0, CUDA 12.6.17
* **Kuantisasi:** 4-bit NormalFloat (NF4) melalui BitsAndBytes (load\_in\_4bit=True). Ini mengurangi penggunaan memori model secara drastis sambil mempertahankan presisi bobot melalui distribusi normal yang dioptimalkan.36
* **Konfigurasi PEFT:** QLoRA dengan Rank () = 16 dan Alpha = 16. Kami menargetkan semua lapisan linier (target\_modules = ["q\_proj", "k\_proj", "v\_proj", "o\_proj", "gate\_proj", "up\_proj", "down\_proj"]).37
* **Arsitektur Target:** Adaptasi lapisan linier penuh ini menghasilkan **28 lapisan yang ditambal (*patched layers*)** dan **18.464.768 parameter yang dapat dilatih**, yang merupakan 1,78% dari total 1.036.449.280 parameter model.17
* **Jendela Konteks:** Panjang urutan maksimum (**Max Sequence Length**) ditetapkan pada **2048 token**.17
* **Presisi:** *Automatic Mixed Precision* (AMP) dengan dtype=None (secara otomatis mendeteksi Float16 pada T4).
* **Optimasi Memori:** Penggunaan Unsloth memberikan percepatan **2x** dalam *throughput* pelatihan dan pengurangan memori yang signifikan 13, memungkinkan ukuran *batch* yang lebih besar yang sebaliknya akan menyebabkan kesalahan *Out-of-Memory* (OOM) pada GPU T4.

### 4.4 Inferensi dan Logika Chain-of-Verification

HexaSLM menggunakan templat prompt terstruktur untuk memicu perilaku CoVe yang dipelajari selama pelatihan. Logika inferensi 17 dikodekan secara keras untuk menegakkan format respons terstruktur:

<|im\_start|>system

You are a cybersecurity expert. Verify all advice systematically.<|im\_end|>

<|im\_start|>user

{question}<|im\_end|>

<|im\_start|>assistant

Let me provide thoroughly verified cybersecurity guidance.

**Step 1 - Initial Analysis:**

*Pre-fill* ini memaksa model untuk memulai generasinya dengan fase analitis alih-alih langsung melompat ke kesimpulan. Parameter generasi (temperature=0.7, top\_p=0.9, repetition\_penalty=1.1) 17 disetel untuk menyeimbangkan kreativitas yang diperlukan untuk pemodelan ancaman dengan determinisme yang diperlukan untuk akurasi teknis. Penalti pengulangan (1.1) sangat penting untuk mencegah model terjebak dalam loop verifikasi yang berulang.

## 5. Hasil Eksperimental

Evaluasi HexaSLM dilakukan menggunakan set pengujian komprehensif yang terdiri dari **400 pertanyaan** yang mencakup empat kategori utama: **OWASP Top 10**, **Panduan NIST**, **Kerentanan Umum**, dan **Praktik Terbaik**.17 Metrik evaluasi difokuskan pada Akurasi, Tingkat Halusinasi, dan Kepatuhan CoVe.

### 5.1 Kinerja Kuantitatif

HexaSLM secara signifikan mengungguli semua model baseline, termasuk model dasar Qwen 2.5 dan pendekatan *fine-tuning* standar.

Tabel 1: Perbandingan Hasil Utama 17

| **Model** | **Akurasi (%)** | **Halusinasi (%)** | **Kepatuhan CoVe (%)** | **F1-Score** |
| --- | --- | --- | --- | --- |
| Qwen2.5-1.5B (Base) | 65,2 | 42,1 | 0,0 | 0,623 |
| Qwen2.5 + Single-stage FT | 72,8 | 28,3 | 15,2 | 0,701 |
| Qwen2.5 + Standard LoRA | 74,5 | 25,7 | 8,1 | 0,718 |
| **HexaSLM (Ours)** | **82,0** | **14,0** | **86,0** | **0,901** |

Seperti yang ditunjukkan pada **Tabel 1**, HexaSLM mencapai **akurasi 82,0%**, peningkatan sebesar **16,8 poin persentase** dibandingkan model dasar. Lebih penting lagi, **tingkat halusinasi turun hampir dua pertiga**, dari 42,1% menjadi 14,0%. Hasil ini sangat mendukung hipotesis kami bahwa CoVe, ketika "dipanggang" ke dalam bobot model melalui *fine-tuning*, bertindak sebagai regularisasi yang kuat terhadap fabrikasi. Skor **Kepatuhan CoVe** sebesar **86,0%** mengonfirmasi bahwa model berhasil menginternalisasi struktur verifikasi multi-langkah, tidak seperti model *Single-stage FT* yang hanya mematuhi 15,2% dari waktu. Skor F1 sebesar 0,901 menunjukkan keseimbangan yang sangat baik antara presisi dan *recall* dalam memberikan saran keamanan yang benar.

### 5.2 Analisis Spesifik Kategori

Untuk memahami kekuatan dan kelemahan model, kami menganalisis kinerja di berbagai domain keamanan.

Tabel 2: Metrik Per-Kategori 17

| **Kategori** | **Akurasi (%)** | **Halusinasi (%)** | **Kepatuhan CoVe (%)** |
| --- | --- | --- | --- |
| **OWASP Top 10** | **88,0** | 16,0 | 80,0 |
| **Praktik Terbaik** | 84,0 | **8,0** | **96,0** |
| **Panduan NIST** | 84,0 | 20,0 | 80,0 |
| **Kerentanan Umum** | 72,0 | 12,0 | 88,0 |

* **OWASP Top 10:** Model berkinerja terbaik di sini (akurasi 88%), kemungkinan besar karena representasi konsep keamanan web yang tinggi dalam data pelatihan.
* **Praktik Terbaik (*Best Practices*):** Kategori ini melihat tingkat halusinasi terendah (8,0%) dan kepatuhan CoVe tertinggi (96,0%). Pertanyaan dalam kategori ini (misalnya, "Bagaimana cara mengonfigurasi header keamanan?") umumnya memiliki jawaban definitif yang diterima secara luas, yang selaras dengan pelatihan model.
* **Panduan NIST:** Meskipun akurasinya tinggi (84,0%), kategori ini memiliki tingkat halusinasi tertinggi (20,0%). Kami berhipotesis bahwa ini disebabkan oleh sifat dokumen NIST yang padat dan bersifat regulasi (misalnya21), di mana model mungkin berhalusinasi mengenai nomor publikasi spesifik atau pengidentifikasi sub-kontrol meskipun konsep umumnya benar.
* **Kerentanan Umum:** Akurasi yang lebih rendah (72,0%) menunjukkan model sedikit berjuang dengan nuansa CVE tertentu atau mekanika eksploitasi yang sangat spesifik dibandingkan dengan kerangka kerja yang lebih luas.

### 5.3 Studi Kasus Kualitatif

Efikasi HexaSLM paling baik dipahami melalui contoh spesifik dari output "Panduan Terverifikasi"-nya.17

#### Studi Kasus 1: Pencegahan Cross-Site Request Forgery (CSRF)

Ketika ditanya tentang pencegahan CSRF, model tidak hanya mencantumkan "gunakan token". Sebaliknya, ia memulai fase **Perencanaan Verifikasi** 17:

* *Pemeriksaan Kritis Q1:* "Apakah ini akurat secara teknis menurut standar OWASP/NIST/CIS?"
* *Pemeriksaan Kritis Q3:* "Apakah ini menjaga batasan etika?"

Respons akhir memberikan rencana 9 poin yang komprehensif termasuk **Validasi Berbasis Token**, **Atribut Cookie SameSite**, dan **Token Berbasis Interaksi**.17 Langkah verifikasi eksplisit "Tidak ada pelanggaran kebijakan privasi yang terdeteksi" 17 memastikan saran tersebut tidak hanya sehat secara teknis tetapi juga aman secara operasional. Model ini tidak terjebak dalam loop pengulangan dan memberikan detail implementasi yang spesifik seperti header X-CSRF-TOKEN dan arahan Content Security Policy.

#### Studi Kasus 2: Injeksi SQL di Python Flask

Untuk kueri "Bagaimana cara mencegah injeksi SQL di Python Flask?", model memberikan cuplikan kode menggunakan ORM Flask-SQLAlchemy. Menariknya, evaluasi menandai respons ini memiliki halusinasi (has\_hallucination: True) tetapi tetap menandainya is\_correct: True.17 Setelah tinjauan manual, perbedaan ini kemungkinan muncul dari model yang menghalusinasikan impor pustaka spesifik yang tidak kritis atau komentar yang tidak memengaruhi keamanan fungsional kode. Ini menyoroti nuansa "halusinasi" dalam tugas pengkodean—solusi yang secara fungsional benar masih dapat mengandung fabrikasi kecil.

## 6. Studi Ablasi

### 6.1 Dampak Chain-of-Verification

Membandingkan "Single-stage FT" (instruction tuning tanpa struktur CoVe) dengan HexaSLM mengungkapkan dampak terisolasi dari mekanisme CoVe. Model satu tahap mencapai akurasi 72,8% dengan halusinasi 28,3%. Dengan menegakkan struktur CoVe (Draft -> Plan -> Verify), akurasi melonjak menjadi 82,0% dan halusinasi turun menjadi 14,0%.17 Ini menegaskan bahwa *struktur* generasi sama pentingnya dengan *konten*. Langkah-langkah CoVe memaksa model untuk "memperhatikan" penalarannya sendiri, menangkap kesalahan dalam fase Initial Analysis sebelum kesalahan tersebut merambat ke panduan akhir.

### 6.2 Peran PRM800K

Penggabungan PRM800K 14 sangat penting. Eksperimen awal (tidak dirinci dalam Tabel 1) tanpa data penalaran menunjukkan model berjuang untuk mempertahankan format CoVe pada jendela konteks yang panjang. Logika langkah-demi-langkah yang melekat dalam dataset matematika tampaknya mentransfer ke logika verifikasi keamanan langkah-demi-langkah, memungkinkan model untuk mempertahankan koherensi selama langkah-langkah "Perencanaan Verifikasi" yang kompleks (misalnya, memeriksa kasus tepi untuk serangan Deserialisasi 17).

### 6.3 Efisiensi Unsloth & QLoRA

Melatih model 1,5B hanya pada dua Tesla T4 akan sangat lambat secara tidak praktis dengan pelatihan FP32 standar. Penggunaan **Unsloth** dan **QLoRA** memungkinkan kami untuk memuat model dan gradien ke dalam kurang dari 15GB VRAM 17, dengan pelatihan selesai dalam sebagian kecil waktu yang dibutuhkan untuk implementasi HuggingFace standar. Efisiensi ini sangat penting bagi komunitas keamanan siber, karena memungkinkan organisasi untuk sering melatih ulang model pada intelijen ancaman terbaru (misalnya, eksploitasi *zero-day* baru) tanpa biaya cloud yang besar.

## 7. Diskusi

### 7.1 Implikasi untuk Keamanan Otomatis

Hasil ini menunjukkan bahwa **Small Language Models (SLM)** seperti HexaSLM dapat memberikan kinerja jauh di atas kelas beratnya ketika dilatih dengan struktur verifikasi yang ketat. Untuk SOC, ini berarti model 1,5B yang diterapkan secara lokal dapat berfungsi sebagai asisten analis "Tingkat 1" yang andal, menyaring peringatan dan menyarankan remediasi dengan tingkat kesalahan yang diketahui dan dapat diterima (14%), yang secara signifikan lebih rendah daripada model generik (42,1%).

### 7.2 Tantangan Halusinasi NIST

Tingkat halusinasi yang lebih tinggi dalam kategori NIST (20%) 17 menunjukkan keterbatasan dalam kemampuan model untuk menghafal fakta hafalan (seperti nomor kontrol SP 800-53 tertentu) dibandingkan dengan penalaran konseptual. Ini menunjukkan bahwa untuk tugas kepatuhan regulasi, lapisan **Retrieval-Augmented Generation (RAG)** 26 mungkin masih diperlukan untuk melengkapi penalaran intrinsik CoVe. Pengetahuan intrinsik cukup untuk *konsep*, tetapi alat ekstrinsik diperlukan untuk *sitasi* dokumen hukum.

### 7.3 Pertimbangan Etika dan Keselamatan

HexaSLM secara eksplisit dilatih untuk menolak instruksi berbahaya. Evaluasi mengonfirmasi skor tinggi dalam **Pemeriksaan Etis** (89,0%).17 Sebagai contoh, dataset memastikan model memberikan saran *defense-only* untuk eksploitasi seperti XSS, berfokus pada mitigasi daripada persenjataan.16 Penyelarasan ini tidak dapat ditawar untuk teknologi penggunaan ganda (*dual-use technologies*) dalam keamanan siber. Dalam tabel evaluasi kualitatif, model secara konsisten memeriksa "Apakah ini menjaga batasan etika (misalnya, tidak ada phishing atau ransomware)?" sebelum menghasilkan output.17

## 8. Kesimpulan

HexaSLM mendemonstrasikan bahwa sifat "kotak hitam" dari LLM dapat diterangi dan dijinakkan melalui inovasi struktural seperti *Chain-of-Verification*. Dengan melakukan *fine-tuning* pada model Qwen 2.5 yang ringkas pada kurikulum pengawasan proses dan pengetahuan domain, kami mencapai asisten keamanan yang sangat akurat dan berorientasi pada verifikasi. Dengan F1-score 0,901 dan tingkat halusinasi hanya 14%, HexaSLM mewakili langkah signifikan menuju AI yang tepercaya dalam keamanan siber.

Pekerjaan di masa depan akan berfokus pada pengintegrasian RAG untuk mengatasi kesenjangan halusinasi NIST dan memperluas metodologi CoVe ke alur kerja "Agentic" di mana langkah verifikasi mencakup eksekusi kode aktual (misalnya, menjalankan pemindaian Nmap untuk memverifikasi kerentanan). Kami mengundang komunitas untuk membangun di atas pekerjaan ini menggunakan repositori yang dirilis.

**Repositori:**

* **Kode:** <https://github.com/AneKazek/HexaSLM>
* **Model:** <https://huggingface.co/anekazek/hexaslm-qwen2.5-cybersec-cove>

## 9. Analisis Mendalam Hasil dan Metodologi

### 9.1 Dekonstruksi Dinamika Pelatihan

Pilihan **Qwen 2.5-1.5B-Instruct** sebagai model dasar didorong oleh kemampuan "Code-Switching" dan dasar penalaran logis yang kuat.9 Sementara model yang lebih besar (7B, 70B) umumnya menawarkan kinerja yang lebih unggul, kelas 1,5B mewakili "titik manis" (*sweet spot*) untuk penyebaran *edge*—kritis untuk agen keamanan yang harus berjalan di laptop analis atau di dalam *enclave* yang terisolasi dari udara (*air-gapped*).

Proses pelatihan menggunakan kernel yang dioptimalkan **Unsloth** 12, yang memfusikan operasi seperti RoPE (*Rotary Positional Embeddings*) dan RMSNorm untuk meminimalkan penggunaan *bandwidth* memori. Ini memungkinkan kami mempertahankan jendela konteks **2048 token**, yang penting untuk memproses log keamanan atau cuplikan kode yang panjang. Konfigurasi **QLoRA** menargetkan semua lapisan linier (Q, K, V, O, MLP) 17, memastikan bahwa adaptasi cukup dalam untuk mengubah gaya penalaran model, bukan hanya kosakata permukaan. Adaptasi "Full-Linear" ini adalah faktor kunci dalam skor kepatuhan CoVe yang tinggi; LoRA standar seringkali hanya menargetkan lapisan *Attention*, yang mungkin tidak cukup untuk mengubah format output struktural model.

### 9.2 Anomali "Halusinasi" dalam Data NIST

Pandangan lebih dalam pada kinerja **Panduan NIST** (Akurasi 84%, Halusinasi 20%) mengungkapkan tren yang menarik. Model sangat akurat dalam *konsep* (misalnya, menjelaskan tujuan Kontrol Akses dengan benar), tetapi rentan terhadap *kesalahan sitasi* (misalnya, mengatribusikan kontrol ke revisi NIST SP 800-53 yang salah). "Halusinasi sitasi" ini adalah efek samping yang diketahui dari pelatihan pada dataset campuran di mana nomor versi (Rev 4 vs Rev 5) mungkin bertentangan. Untuk model keamanan "Terverifikasi", ini menunjukkan bahwa **pengetahuan intrinsik** cukup untuk *konsep* tetapi **alat ekstrinsik** (RAG) diperlukan untuk *sitasi*. Perbedaan ini sangat penting bagi pengguna: percayai HexaSLM untuk *cara* mengamankan sistem, tetapi verifikasi *peraturan mana* yang memandatkannya.

### 9.3 Chain-of-Verification: Penilaian Kualitatif

Contoh kualitatif 17 menunjukkan bahwa CoVe bertindak sebagai "rem kognitif". Dalam respons **Insecure Deserialization**, model berhenti sejenak untuk memeriksa "kasus tepi atau pertimbangan spesifik lingkungan" sebelum menjawab. Langkah refleksif diri ini secara efektif memangkas saran yang tidak aman (misalnya, menggunakan pickle di Python dengan pengamanan yang tidak memadai) yang mungkin dihasilkan oleh model standar karena prevalensi contoh kode yang tidak aman dalam data pra-pelatihan. "Status Verifikasi" di akhir respons (All security checks passed) memberikan sinyal kepercayaan yang ramah pengguna, meniru alur kerja analis senior manusia yang meninjau pekerjaan junior.

### 9.4 Benchmarking Terhadap State of the Art

Dibandingkan dengan lanskap LLM keamanan yang lebih luas (misalnya, SecureBERT 18, Owl, dll.), kontribusi unik HexaSLM adalah **loop verifikasi waktu-inferensi** (*inference-time verification loop*). Sebagian besar LLM keamanan adalah generator "one-shot". HexaSLM adalah pemikir "multi-turn" yang dikemas dalam satu jalur generasi. Skor F1 0,901 17 menunjukkan bahwa pendekatan ini menghasilkan keseimbangan presisi dan *recall* yang jauh lebih baik (menghindari positif palsu/halusinasi dan mencakup semua kontrol keamanan yang diperlukan) daripada *fine-tuning* standar. Peningkatan dibandingkan baseline "Standard LoRA" (F1 0,718) mengukur nilai struktur CoVe itu sendiri—menambahkan "langkah berpikir" bernilai hampir 20 poin dalam kinerja F1.

**Sitasi yang digunakan dalam laporan ini:** .1

#### Works cited

1. LLMs for Cybersecurity in the Big Data Era: A Comprehensive Review of Applications, Challenges, and Future Directions - MDPI, accessed on February 17, 2026, <https://www.mdpi.com/2078-2489/16/11/957>
2. Model Science: getting serious about verification, explanation and control of AI systems, accessed on February 17, 2026, <https://arxiv.org/html/2508.20040v1>
3. CTIBench: A Benchmark for Evaluating LLMs in Cyber Threat Intelligence - arXiv, accessed on February 17, 2026, <https://arxiv.org/html/2406.07599v3>
4. A Comparative Evaluation of Large Language Models in Vulnerability Detection - Network and Distributed System Security (NDSS) Symposium, accessed on February 17, 2026, <https://www.ndss-symposium.org/wp-content/uploads/2025-1491-paper.pdf>
5. Using LLMs for Security Advisory Investigations: How Far Are We? - arXiv, accessed on February 17, 2026, <https://arxiv.org/html/2506.13161v1>
6. Using LLMs for Security Advisory Investigations: How Far Are We? - ResearchGate, accessed on February 17, 2026, <https://www.researchgate.net/publication/392735674_Using_LLMs_for_Security_Advisory_Investigations_How_Far_Are_We>
7. Chain-of-Verification Reduces Hallucination in Large Language Models - ETH Zurich Research Collection, accessed on February 17, 2026, <https://www.research-collection.ethz.ch/server/api/core/bitstreams/468e77de-b21f-4ede-b179-8a52b01a1c5a/content>
8. Chain-of-Verification Reduces Hallucination in Large Language Models - arXiv.org, accessed on February 17, 2026, <https://arxiv.org/abs/2309.11495>
9. [2412.15115] Qwen2.5 Technical Report - arXiv.org, accessed on February 17, 2026, <https://arxiv.org/abs/2412.15115>
10. [2305.14314] QLoRA: Efficient Finetuning of Quantized LLMs - arXiv, accessed on February 17, 2026, <https://arxiv.org/abs/2305.14314>
11. QLoRA vs LoRA: Which Fine‑Tuning Wins? | newline, accessed on February 17, 2026, <https://www.newline.co/@Dipen/qlora-vs-lora-which-finetuning-wins--683ca660>
12. Fine-tuning LLMs Guide | Unsloth Documentation, accessed on February 17, 2026, <https://unsloth.ai/docs/get-started/fine-tuning-llms-guide>
13. [2601.02609] Chronicals: A High-Performance Framework for LLM Fine-Tuning with 3.51x Speedup over Unsloth - arXiv, accessed on February 17, 2026, <https://arxiv.org/abs/2601.02609>
14. openai/prm800k: 800000 step-level correctness labels on LLM solutions to MATH problems, accessed on February 17, 2026, <https://github.com/openai/prm800k>
15. [2305.20050] Let's Verify Step by Step - arXiv, accessed on February 17, 2026, <https://arxiv.org/abs/2305.20050>
16. AlicanKiraz0/Cybersecurity-Dataset-v1 - Hugging Face, accessed on February 17, 2026, <https://huggingface.co/datasets/AlicanKiraz0/Cybersecurity-Dataset-v1>
17. table5\_examples.csv
18. Large Language Model (LLM) for Telecommunications: A Comprehensive Survey on Principles, Key Techniques, and Opportunities, accessed on February 17, 2026, <https://ieeexplore.ieee.org/iel8/9739/11032135/10685369.pdf>
19. Large Language Models in Cybersecurity: State-of-the-Art - SciTePress, accessed on February 17, 2026, <https://www.scitepress.org/Papers/2025/133776/133776.pdf>
20. Large Language Models for Cyber Security: A Systematic Literature Review - arXiv, accessed on February 17, 2026, <https://arxiv.org/html/2405.04760v4>
21. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile | NIST, accessed on February 17, 2026, <https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence>
22. AI Risk Management Framework - NIST, accessed on February 17, 2026, <https://www.nist.gov/itl/ai-risk-management-framework>
23. OWASP Top 10 for Large Language Model Applications, accessed on February 17, 2026, <https://owasp.org/www-project-top-10-for-large-language-model-applications/>
24. OWASP Top 10 Risks for Large Language Models: 2025 updates - Barracuda Blog, accessed on February 17, 2026, <https://blog.barracuda.com/2024/11/20/owasp-top-10-risks-large-language-models-2025-updates>
25. Policy Frameworks for Transparent Chain-of-Thought Reasoning in Large Language Models - arXiv, accessed on February 17, 2026, <https://arxiv.org/html/2503.14521v1>
26. The LLM Data Auditor: A Metric-oriented Survey on Quality and Trustworthiness in Evaluating Synthetic Data - arXiv, accessed on February 17, 2026, <https://arxiv.org/html/2601.17717v1>
27. A Systematic Literature Review of Code Hallucinations in LLMs: Characterization, Mitigation Methods, Challenges, and Future Directions for Reliable AI | Request PDF - ResearchGate, accessed on February 17, 2026, <https://www.researchgate.net/publication/397231387_A_Systematic_Literature_Review_of_Code_Hallucinations_in_LLMs_Characterization_Mitigation_Methods_Challenges_and_Future_Directions_for_Reliable_AI>
28. Let's Verify Step by Step | OpenAI, accessed on February 17, 2026, <https://cdn.openai.com/improving-mathematical-reasoning-with-process-supervision/Lets_Verify_Step_by_Step.pdf>
29. [2410.21228] LoRA vs Full Fine-tuning: An Illusion of Equivalence - arXiv.org, accessed on February 17, 2026, <https://arxiv.org/abs/2410.21228>
30. LoRA vs. QLoRA: Efficient fine-tuning techniques for LLMs - Modal, accessed on February 17, 2026, <https://modal.com/blog/lora-qlora>
31. Unsloth Docs | Unsloth Documentation, accessed on February 17, 2026, <https://unsloth.ai/docs>
32. Unsloth: The Open-Source Library That's Supercharging LLM Fine-Tuning in 2025, accessed on February 17, 2026, <https://www.gocodeo.com/post/unsloth-the-open-source-library-thats-supercharging-llm-fine-tuning>
33. Qwen2.5: A Party of Foundation Models! | Qwen, accessed on February 17, 2026, <https://qwenlm.github.io/blog/qwen2.5/>
34. Qwen2.5-1M Technical Report, accessed on February 17, 2026, <https://qianwen-res.oss-cn-beijing.aliyuncs.com/Qwen2.5-1M/Qwen2_5_1M_Technical_Report.pdf>
35. AlicanKiraz0/Cybersecurity-Dataset-v1 - Hugging Face, accessed on February 17, 2026, <https://huggingface.co/datasets/AlicanKiraz0/Cybersecurity-Dataset-v1/viewer/default/train>
36. QLoRA Fine-Tuning with Unsloth: A Complete Guide - Medium, accessed on February 17, 2026, <https://medium.com/@matteo28/qlora-fine-tuning-with-unsloth-a-complete-guide-8652c9c7edb3>
37. Medical LLMs: Fine-Tuning vs. Retrieval-Augmented Generation - PMC, accessed on February 17, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC12292519/>
38. Artificial Intelligence Risk Management Framework - https: //site.unibo., accessed on February 17, 2026, <https://site.unibo.it/hypermodelex/en/publications/2024-04-01-nist-ai-risk-management-genai.pdf/@@download/file/2024-04-01-NIST-AI-RISK-MANAGEMENT-GENAI.pdf>
39. unslothai/unsloth: Fine-tuning & Reinforcement Learning for LLMs. Train OpenAI gpt-oss, DeepSeek, Qwen, Llama, Gemma, TTS 2x faster with 70% less VRAM. - GitHub, accessed on February 17, 2026, <https://github.com/unslothai/unsloth>
40. Qwen2.5 Technical Report - BibBase, accessed on February 17, 2026, <https://bibbase.org/network/publication/qwen-yang-yang-zhang-hui-zheng-yu-li-etal-qwen25technicalreport-2025>
41. QLoRA: Efficient Finetuning of Quantized LLMs citation - Ruofei Du, accessed on February 17, 2026, <https://duruofei.com/cites/Dettmers2023QLoRA.html>