INTEGRASI KECERDASAN BUATAN GENERATIF UNTUK ANALISIS DAN MITIGASI DATA CVE

Disusun Oleh:

N a m a : Pradipta Putra Abimata
NIM : 20523004

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2024

HALAMAN PENGESAHAN DOSEN PEMBIMBING

INTEGRASI KECERDASAN BUATAN GENERATIF UNTUK ANALISIS DAN MITIGASI DATA CVE

TUGAS AKHIR

Disusun Oleh:

N a m a : Pradipta Putra Abimata
NIM : 20523004

Yogyakarta, 9 Juli 2024
Pembimbing,

(Mukhammad Andri Setiawan S.T., M.Sc., Ph.D.)

ACC Pendadaran

HALAMAN PENGESAHAN DOSEN PENGUJI

INTEGRASI KECERDASAN BUATAN GENERATIF UNTUK ANALISIS DAN MITIGASI DATA CVE

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 19 Juli 2024

Tim Penguji

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D.

Anggota 1

Erika Ramadhani, S.T., M.Eng.

Anggota 2

Rahadian Kurniawan, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.d.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Pradipta Putra Abimata

NIM : 20523004

Tugas akhir dengan judul:

INTEGRASI KECERDASAN BUATAN GENERATIF UNTUK ANALISIS DAN MITIGASI DATA CVE

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 9 Juli 2024

(Pradipta Putra Abimata)

HALAMAN PERSEMBAHAN

Alhamdulillahi Rabbil ‘Alamin. Segala puji bagi Allah Yang Maha Esa yang telah memberikan karunia serta petunjuk-Nya, sehingga kita masih diberi kesempatan untuk hidup dan tetap berada di jalan yang benar menurut agama Islam. Semoga shalawat serta salam senantiasa tercurah kepada Nabi Muhammad shallallahu ‘alaihi wasallam, yang telah membimbing umatnya menuju kehidupan yang lebih baik setelah berada dalam kegelapan.

Penulis berterima kasih kepada Allah Subhanahu wa Ta’ala atas segala nikmat, rahmat, dan ridho-Nya yang tak pernah putus. Shalawat serta salam selalu tercurah kepada Nabi Muhammad SAW, yang menjadi teladan bagi seluruh umatnya.

Terima kasih juga kepada kedua orang tua tercinta, Papah Agung Wijanarko, S.Sos., M.M dan Mamah Vinta Prasasti Dewi
Penulis juga berterima kasih kepada saudara-saudara, yakni kedua saudara Laki - Laki, Yoga Putra Aditama dan Indra Putra Adinata.

Terima kasih juga kepada Teman teman penulis, yang sudah selalu memberi dukungan dan semangat untuk menyelesaikan skripsi, yakni Akmal Zaidan, Gilang Aries, Raffry Rizqullah, Hawada Alfikri, Alber Derry, Daffa Sahhad, Dan teman teman lain yg tidak dapat saya sebutkan satu persatu.

Semoga karya ini bisa
menjadi bentuk bakti dari seorang anak kepada orang tuanya, serta bentuk pertanggungjawaban untuk menyelesaikan pendidikan Sarjana.

HALAMAN MOTO

“Do It for The Plot, the conscious decision to see yourself as the main character of the story that is your life. You maintain the outlook that every moment - good or bad - is merely a plot point for your larger narrative. You are the writer, producer, director, and star of your life. Start living unapologetically and give them a plot twist that no one saw coming.”

KATA PENGANTAR

Terima kasih kepada Allah SWT atas kebaikan, hidayah, dan rahmat-Nya, yang memungkinkan penulis menyelesaikan tugas akhir berjudul "Integrasi Kecerdasan Buatan Generatif Untuk Analisis Dan Mitigasi Data CVE". Kami mengirimkan salam hangat dan doa kami kepada Nabi Muhammad SAW, yang merupakan sumber inspirasi, kebaikan, dan berkah bagi semua umatnya.

Tujuan pembuatan skripsi ini adalah untuk menjadikan lulusan informatika dengan gelar Sarjana Komputer. Terlepas dari semua kendala, penulis mengakui bahwa bantuan, dukungan, dan upaya banyak orang diperlukan untuk menyelesaikan produk akhir ini. Dengan demikian, penulis ingin menggunakan kesempatan ini untuk menyampaikan rasa terima kasih yang tulus kepada:

1. Allah Subhanahu wa Ta'ala atas segala Rahmat, nikmat, serta ridho-Nya sehingga tugas akhir ini dapat terselesaikan dengan baik.
2. Mukhammad Andri Setiawan S.T., M.Sc., Ph.D. selaku dosen pembimbing yang telah memberikan bimbingan, pengarahan, dan arahan dalam menyelesaikan tugas akhir ini.
3. Mukhammad Andri Setiawan S.T., M.Sc., Ph.D. selaku dosen pembimbing akademik yang selalu memberi arahan dan solusi disetiap masalah yang penulis hadapi di perkuliahan.
4. Kedua Orang Tua saya tersayang, Papah Agung Wijanarko dan Mamah Vinta Prasasti Dewi yang selalu memberikan dukungan dalam bentuk apapun, semangat, materi, waktu, dan kasih sayang.
5. Serta masih banyak lagi orang orang yang tidak bisa saya sebutkan nama nya satu per satu, yang pernah terlibat untuk membantu pengerjaan penelitian.

Umumnya Alhamdulillah, puji syukur kami panjatkan ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga skripsi yang berjudul "Integrasi Kecerdasan Buatan Generatif untuk Analisis dan Mitigasi Data CVE" ini dapat diselesaikan. Semoga Allah terus melimpahkan penulis dengan kebaikan dan sukacita atas semua bantuan-Nya.

Dengan selesainya tugas akhir ini, penulis bertujuan untuk berkontribusi dalam bidang teknologi informasi dan meningkatkan keamanan serta mitigasi kerentanan dalam sistem informasi melalui analisis data CVE. Pada saat tulisan ini selesai, penulis menemukan bahwa produk jadi masih belum sempurna. Akibatnya, kritik dan rekomendasi yang bermanfaat sangat

disambut baik untuk meningkatkan produk akhir ini ke depan. Dengan kerendahan hati, penulis ingin mengucapkan terima kasih kepada semua orang yang membantu sepanjang jalan dalam menulis produk akhir ini.

Semoga setiap perbaikan dan peningkatan yang diimplementasikan dapat membawa manfaat yang lebih besar. Keterlibatan, dukungan, dan pemikiran konstruktif dari semua pihak menjadi pendorong utama kesuksesan penyelesaian tugas akhir ini. Penulis mengharapkan agar semangat kolaborasi ini terus terjaga, dan penelitian berkelanjutan dapat memberikan kontribusi yang lebih signifikan bagi pengembangan ilmu pengetahuan dan praktik di masa depan.

Dengan kerendahan hati dan penuh rasa syukur, penulis mengakui semua bantuan dan peluang yang telah diterimanya. Semoga tugas akhir ini tidak hanya menjadi penutup bab di masa perkuliahan, tetapi juga awal dari perjalanan baru dalam berkontribusi bagi kemajuan ilmu dan masyarakat.

Yogyakarta, 9 Juli 2024

(Pradipta Putra Abimata)

SARI

Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi berbasis Flask yang mampu melakukan analisis terhadap Common Vulnerabilities and Exposures (CVE) dengan memanfaatkan kecerdasan buatan generatif dari OpenAI. Latar belakang dari penelitian ini adalah kebutuhan yang semakin mendesak akan alat yang mampu menganalisis dan memberikan mitigasi atas kerentanan keamanan siber secara cepat dan akurat. Dalam era digital saat ini, CVE menjadi salah satu referensi utama untuk mengetahui kelemahan sistem keamanan yang ada, sehingga pemahaman yang mendalam tentang CVE sangat penting bagi para praktisi keamanan siber.

Gambaran singkat penelitian ini meliputi pengembangan aplikasi web yang terintegrasi dengan OpenAI API untuk memberikan analisis CVE yang komprehensif. Aplikasi ini tidak hanya menampilkan informasi dasar tentang CVE, tetapi juga memberikan analisis mendetail tentang dampak, vektor serangan, dan langkah-langkah mitigasi yang direkomendasikan. Dalam proses pengembangan, aplikasi ini menggunakan framework Flask di sisi backend dan Tailwind CSS di sisi frontend untuk memastikan aplikasi memiliki antarmuka yang ramah pengguna dan mudah dioperasikan.

Metodologi yang digunakan dalam penelitian ini mencakup beberapa tahap, mulai dari pengumpulan data CVE terbaru, perancangan dan pengembangan aplikasi, integrasi dengan OpenAI API, hingga pengujian aplikasi. Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memberikan hasil analisis yang akurat dan relevan. Proses pengujian melibatkan evaluasi terhadap kemampuan aplikasi dalam mengambil data CVE, menganalisisnya, dan memberikan rekomendasi mitigasi.

Temuan utama dari penelitian ini menunjukkan bahwa aplikasi yang dikembangkan mampu memberikan analisis yang komprehensif dan akurat terhadap CVE. Hasil pengujian menunjukkan bahwa aplikasi dapat menghasilkan informasi yang mendetail tentang CVE, dampaknya, dan langkah-langkah mitigasi yang perlu diambil. Dengan demikian, aplikasi ini diharapkan dapat menjadi alat yang berguna bagi para praktisi keamanan siber dalam mengidentifikasi dan mengatasi kerentanan keamanan dengan lebih efisien.

Kata kunci: CVE, analisis CVE, mitigasi kerentanan, kecerdasan buatan, OpenAI, Flask, keamanan siber

GLOSARIUM

Glosarium memuat daftar kata tertentu yang digunakan dalam laporan dan membutuhkan penjelasan, misalnya kata serapan yang belum lazim digunakan. Urutkan sesuai abjad. Contoh penulisannya seperti di bawah ini:

Analisis CVE	Proses penilaian dan evaluasi terhadap kerentanan yang terdaftar dalam Common Vulnerabilities and Exposures (CVE).
Backend	Bagian dari aplikasi yang berjalan di server dan bertanggung jawab untuk logika bisnis, pengelolaan database dan pemrosesan permintaan frontend.
CVE	Common Vulnerabilities and Exposures adalah sistem referensi umum untuk mengidentifikasi dan mendaftar kerentanan keamanan dalam perangkat lunak dan sistem komputer.
Frontend	Bagian dari aplikasi yang berinteraksi langsung dengan pengguna, bertanggung jawab untuk antarmuka pengguna dan pengiriman permintaan ke backend.
HTTP	Protokol yang digunakan untuk mengirim dan menerima informasi di web, sering digunakan untuk komunikasi antara web browser dan server.
Kecerdasan Buatan	Teknologi yang memungkinkan mesin untuk melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia, seperti pengenalan suara, pengambilan keputusan, dan analisis data.
OpenAI API	Antarmuka pemrograman aplikasi yang disediakan oleh OpenAI untuk memungkinkan pengembang mengakses dan menggunakan model AI yang dikembangkan oleh OpenAI.
JSON	Format pertukaran data yang mudah dibaca oleh manusia dan mudah diparsing oleh mesin, digunakan secara luas untuk komunikasi antara server dan web aplikasi.
Prompt Engineering	Proses merancang dan menyusun input yang diberikan kepada model AI untuk mendapatkan respons yang diinginkan, termasuk penggunaan teknik khusus untuk memaksimalkan performa model.

Role-Prompting	Teknik dalam prompt engineering di mana model AI diberikan peran tertentu untuk memandu respons yang dihasilkan agar sesuai dengan konteks yang diinginkan.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir, perkembangan pesat dalam teknologi informasi dan komunikasi telah mengubah cara organisasi beroperasi. Transformasi digital ini telah meningkatkan efisiensi dan produktivitas, namun juga membawa ancaman baru berupa serangan siber. Data terbaru menunjukkan bahwa pada tahun 2023, lebih dari 28.000 CVE baru telah diterbitkan, mencerminkan peningkatan yang signifikan dalam jumlah kerentanan yang terdeteksi dibandingkan tahun-tahun sebelumnya (SecurityWeek, 2023). Kenaikan ini mencerminkan kompleksitas yang meningkat dari lanskap keamanan siber, di mana ancaman menjadi lebih beragam dan serangan semakin sulit untuk dideteksi dan ditanggulangi (Tang dkk., 2023).

Ancaman ini tidak hanya semakin kompleks dan beragam, tetapi juga meningkat dalam frekuensi dan dampaknya. Insiden seperti pencurian data, penyebaran malware, dan serangan ransomware menjadi lebih umum dan merugikan. Dampak dari serangan siber ini tidak hanya terbatas pada kerugian finansial, tetapi juga dapat merusak reputasi organisasi dan menyebabkan hilangnya kepercayaan publik. Dalam konteks ini, penting bagi organisasi untuk mengambil langkah-langkah proaktif dalam mengamankan aset digital mereka.

Untuk mengamankan aset digital dan data sensitif, organisasi di seluruh dunia harus mengadopsi langkah-langkah keamanan yang lebih proaktif dan canggih. Common Vulnerabilities and Exposures (CVE) adalah alat utama yang digunakan untuk mengidentifikasi dan mengklasifikasikan kerentanan keamanan dalam perangkat lunak dan perangkat keras. CVE adalah daftar standar internasional yang dikelola oleh Mitre Corporation dan diakui secara luas oleh industri keamanan siber. Setiap entri CVE menjelaskan kerentanan spesifik yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab (Aghaei dkk., 2023).

Kerentanan CVE sering menjadi target utama bagi penyerang siber karena dapat memberikan akses tidak sah ke sistem, mencuri informasi sensitif, atau merusak integritas data. Oleh karena itu, penting bagi organisasi untuk secara proaktif mengidentifikasi dan mengurangi kerentanan ini sebelum dapat dieksploitasi. Namun, tantangan yang dihadapi adalah volume data CVE yang sangat besar dan terus berkembang, sehingga analisis manual menjadi tidak efisien dan rentan terhadap kesalahan (GM dkk., 2020).

Masalah utama yang dihadapi oleh organisasi dalam konteks ini adalah ketidakmampuan untuk secara efektif memantau, menganalisa, dan menanggapi kerentanan yang terus berkembang. Proses manual tidak hanya memakan waktu dan tenaga, tetapi juga rentan terhadap kesalahan manusia, yang dapat menyebabkan kerentanan tidak terdeteksi dan dieksploitasi oleh penyerang. Menurut (Aghaei dkk., 2023), tantangan ini diperparah oleh kompleksitas dan dinamika ancaman keamanan siber yang terus berevolusi. Selain itu, banyak organisasi tidak memiliki sumber daya atau keahlian yang diperlukan untuk secara efektif menganalisis dan menanggapi ancaman yang teridentifikasi. Dengan jumlah data yang begitu besar dan terus berkembang, menjadi sangat sulit untuk memilah dan mengidentifikasi kerentanan yang paling kritis dan membutuhkan perhatian segera (Guo dkk., 2022).

Untuk mengatasi tantangan ini, teknologi kecerdasan buatan menawarkan solusi yang dapat menawarkan solusi yang dapat membantu dan menganalisa dan mengelola data CVE dengan lebih efisien. Kecerdasan buatan adalah bidang ilmu komputer yang berfokus pada pengembangan sistem yang dapat melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia, seperti pengenalan suara, pemahaman bahasa, pengambilan keputusan, dan pemecahaan masalah (Kilani dkk., 2018). AI dapat dibagi menjadi dua kategori utama yaitu AI sempit (narrow AI), yang dirancang untuk tugas-tugas spesifik, dan AI umum (general AI), yang memiliki kemampuan untuk melakukan berbagai tugas intelektual yang dilakukan oleh manusia (Kulkarni & S. N, 2023). Sejarah AI dimulai pada tahun 1950-an dengan karya Alan Turing, yang mengusulkan 'tes Turing' sebagai ukuran kecerdasan mesin (Jittprasong, 2021). Pada tahun 1956, istilah 'Artificial Intelligence' pertama kali diperkenalkan oleh John McCarthy pada konferensi Dartmouth (Verma & Verma, 2022). Sejak itu, AI telah mengalami berbagai fase perkembangan, mulai dari ekspektasi tinggi dan pendanaan yang melimpah hingga periode yang dikenal sebagai 'musim dingin AI', di mana minat dan investasi menurun. Namun, kemajuan dalam teknologi komputasi dan algoritma pembelajaran mesin pada akhir abad ke-20 dan awal abad ke-21 telah menghidupkan kembali bidang ini (Fradkov & Shepeljavyi, 2022). Saat ini, AI digunakan secara luas dalam berbagai aplikasi, termasuk asisten virtual, kendaraan otonom, analisis data, dan banyak lagi, menunjukkan potensinya untuk merevolusi berbagai aspek kehidupan manusia (Jha dkk., 2023).

Dalam konteks ini, kecerdasan buatan menawarkan solusi potensial yang dapat membantu dalam menganalisis data CVE secara lebih efisien. Teknologi AI generatif dapat digunakan untuk mengidentifikasi pola-pola tertentu dalam data CVE dan menghasilkan rekomendasi mitigasi yang lebih akurat dan efektif. Menurut (Aghaei dkk., 2023), AI generatif

bekerja dengan menggunakan algoritma pembelajaran mesin untuk memproses dan menganalisa data dalam skala besar dengan kecepatan tinggi, mengidentifikasi kerentanan potensial, dan mengusulkan langkah-langkah mitigasi yang tepat. Selain itu, penelitian oleh (Sabeel dkk., 2023) menunjukan bahwa serangan siber dapat dianalisis dan diidentifikasi dengan menggunakan model AI generatif, yang memungkinkan deteksi dan mitigasi ancaman yang lebih efisien.

Proses bisnis yang dapat dioptimalkan melalui integrasi ini meliputi analisis risiko, di mana AI dapat mengevaluasi potensi dampak dari kerentanan yang ditemukan, membantu organisasi memahami risiko yang mereka hadapi. Berdasarkan analisis data, AI generatif dapat memberikan rekomendasi mitigasi yang akurat, seperti patching atau konfigurasi ulang sistem, yang dapat diimplementasikan dengan cepat. Rekomendasi ini membantu mengurangi waktu respon dan memungkinkan organisasi untuk mengatasi kerentanan sebelum dapat dieksploitasi.

Dengan otomatisasi analisis CVE, tim keamanan dapat fokus pada tugas-tugas yang lebih strategis, seperti perencanaan kebijakan keamanan dan pelatihan karyawan, mengurangi beban kerja manual yang monoton dan rentan terhadap kesalahan (Khlaisamniang dkk., 2023). AI generatif juga dapat digunakan untuk memantau terus-menerus data CVE yang baru diterbitkan dan memberikan peringatan dini tentang kerentanan yang mungkin mempengaruhi sistem organisasi, memungkinkan respon yang lebih cepat dan proaktif terhadap ancaman yang muncul. Selain itu, AI generatif dapat membantu dalam mengidentifikasi pola-pola tertentu dalam data CVE yang mungkin terlewatkan oleh analisis manual, memberikan wawasan yang lebih mendalam dan penilaian yang lebih akurat mengenai kerentanan yang ada.

Penelitian ini bertujuan untuk mengkaji bagaimana integrasi AI generatif dapat diimplementasikan dalam analisis dan mitigasi data CVE. Penelitian ini juga akan mengeksplorasi manfaat yang dapat diperoleh, seperti peningkatan efisiensi dan akurasi dalam identifikasi dan mitigasi kerentanan, serta tantangan yang mungkin dihadapi dalam proses implementasi, seperti kebutuhan akan data yang berkualitas tinggi dan infrastruktur teknologi yang memadai (Bajaj & Samal, 2023).

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang akan dibahas dalam skripsi ini yaitu untuk menganalisis dan meringkas informasi kerentanan keamanan secara komprehensif, memberikan rekomendasi mitigasi yang disesuaikan dengan tingkat keahlian pengguna, memungkinkan kolaborasi dan penyimpanan hasil analisis kerentanan antar

pengguna, serta mengoptimalkan pencarian dan integrasi informasi kerentanan dari berbagai sumber.

1.3 Batasan Masalah

Untuk memastikan fokus penelitian ini tetap terarah dan sesuai dengan tujuan yang ingin dicapai, beberapa batasan permasalahan yang disesuaikan dengan kode yang telah dikembangkan adalah sebagai berikut:

a. Ruang Lingkup Data CVE

   Penelitian ini hanya akan menggunakan data Common Vulnerabilities and Exposures (CVE) yang diambil dari National Vulnerability Database (NVD). Pengambilan data CVE dilakukan melalui API yang disediakan oleh NVD.

b. Analisis dan Rekomendasi Mitigasi

   Penelitian ini akan membatasi analisis dan rekomendasi mitigasi hanya pada informasi yang tersedia dari data CVE dan respon yang dihasilkan oleh model AI generatif. Penelitian tidak akan mencakup analisis manual tambahan atau sumber data lain di luar NVD dan OpenAI.

c. Interaksi pengguna

   Aplikasi yang dikembangkan akan mengizinkan pengguna untuk mengajukan pertanyaan terkait CVE, mengunggah file CSV berisi ID CVE. Fokus penelitian adalah pada peningkatan akurasi dalam analisis CVE melalui fitur-fitur ini.

d. Implementasi dan Pengujian

   Implementasi dan pengujian aplikasi akan dibatasi pada lingkungan pengembangan yang menggunakan Flask sebagai kerangka kerja web. Penelitian ini tidak mencakup penerapan pada sistem produksi skala besar atau uji coba pada berbagai platform keamanan siber.

e. Tantangan dan Keterbatasan AI

   Penelitian ini akan mempertimbangkan tantangan yang terkait dengan penggunaan AI generatif, seperti kebutuhan data berkualitas tinggi, latensi dalam respon AI, dan keterbatasan dalam memahami konteks spesifik dari setiap kerentanan.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengeksplorasi bagaimana teknologi AI generatif dapat diimplementasikan dalam analisis dan mitigasi data CVE. Secara khusus penelitian ini bertujuan untuk :

a. Mengidentifikasi dan menganalisis kerentanan CVE menggunakan teknologi AI generatif, serta mengeksplorasi cara-cara yang dapat digunakan untuk mengelola dan memproses data CVE dengan bantuan AI generatif.
b. Memberikan rekomendasi mitigasi untuk kerentanan CVE berdasarkan hasil analisis yang dilakukan oleh AI generatif.
c. Mengembangkan prototipe aplikasi berbasis AI generatif yang dapat digunakan oleh organisasi untuk membantu memprioritaskan dan menangani kerentanan yang paling kritis, serta merancang dan menguji aplikasi ini untuk melihat bagaimana AI generatif dapat digunakan dalam konteks keamanan siber.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan berbagai manfaat baik dari segi akademis maupun praktis, sebagai berikut:

a. Peningkatan Pengelolaan Kerentanan: Dengan menggunakan AI generatif untuk menganalisis dan meringkas informasi kerentanan CVE, penelitian ini dapat membantu organisasi dalam pengelolaan kerentanan yang lebih baik.
b. Pengembangan Rekomendasi Mitigasi: Penelitian ini bertujuan untuk menghasilkan rekomendasi mitigasi berdasarkan analisis data CVE yang dilakukan oleh AI generatif, yang dapat digunakan oleh organisasi dalam menangani kerentanan.
c. Prototipe Aplikasi Berbasis AI: Penelitian ini menghasilkan prototipe aplikasi yang memanfaatkan AI generatif, yang dapat digunakan oleh organisasi untuk membantu dalam prioritas dan penanganan kerentanan yang ada.

1.6 Metodologi Penelitian

Untuk mencapai keberhasilan dalam penelitian ini, digunakan strategi penelitian dan pengembangan yang dipadukan dengan proses pengembangan *Waterfall*. Pendekatan ini melibatkan beberapa tahapan yang diadaptasi dari metode *Waterfall* untuk memastikan penelitian berjalan sistematis dan iteratif. Berikut adalah tahapan metodologi penelitian yang digunakan:

a. Analisis Kebutuhan
    1. Data CVE: Menggunakan data CVE dari National Vulnerability Database (NVD) melalui API.

2. Teknologi AI Generatif: Menggunakan model GPT dari OpenAI untuk analisis dan rekomendasi mitigasi
3. Fitur utama: Mencakup fungsi-fungsi seperti fetch_cve_data, extract_details, get_openai_response, dan perform_web_search.
4. Interaksi Pengguna: Mengizinkan pengguna untuk mengajukan pertanyaan terkait CVE, mengunggah file CSV berisi ID CVE, dan melihat Riwayat percakapan.
5. Autentikasi Pengguna: Menggunakan Flask-Login untuk login, logout, dan pendaftaran pengguna.

b. Desain Sistem

1. Arsitektur Aplikasi: Aplikasi berbasis web menggunakan Flask Sebagai kerangka kerja.
2. Database: Menggunakan SQLAlchemy dengan SQLite untuk penyimpanan data.
3. Antarmuka Pengguna: Desain halaman web untuk *login*, registrasi, Dasbor, *home* dan *library.*
4. Fungsi Utama: Merancang dan mendokumentasikan fungsi-fungsi utama yang akan diimplementasikan.
5. Keamanan: Menyediakan autentikasi dan otorisasi untuk melindungi data pengguna.

c. Implementasi

1. Inisialisasi Flask App: Konfigurasi Flask, SQLAlchemy, dan Flask-Login.
2. Model Database: Implementasi model User, Thread, dan CVEEntry.
3. Rute dan Endpoint: Implementasi berbagai *endpoint* untuk *login*, *logout*, registrasi, pengelolaan thread, dan pengolahan data CVE.
4. Integrasi API: Mengintegrasikan API dari NVD untuk fetch data CVE dan OpenAI untuk analisis dan rekomendasi mitigasi.
5. Proses CSV dan *Web Search*: Implementasi fungsi untuk *upload file* CSV dan pencarian web.

d. Pengujian

Pengujian Fungsional: Pengujian fungsional adalah proses yang krusial dalam pengembangan perangkat lunak, karena bertujuan untuk memastikan bahwa setiap fitur dan

fungsi dalam aplikasi bekerja sesuai dengan yang diharapkan. Dalam penelitian ini, pengujian dilakukan dengan menggunakan metode black box testing, yang menitikberatkan pada pengujian antarmuka pengguna dan hasil keluaran aplikasi tanpa memperhatikan struktur internal atau kode sumber. Black box testing memungkinkan penguji untuk memverifikasi bahwa input yang diberikan menghasilkan output yang diinginkan, sesuai dengan spesifikasi fungsional yang telah ditentukan. Pada aplikasi ini, pengujian fungsional mencakup berbagai skenario penggunaan, termasuk pengujian terhadap setiap endpoint dan fungsionalitas utama.

e. Pemeliharaan
    1. Perbaikan *Bug*: Memperbaiki bug yang ditemukan setelah *deployment*.
    2. Peningkatan Fitur: Menambahkan atau meningkatkan fitur berdasarkan masukan pengguna.
    3. Pemantauan: Memantau performa dan keamanan aplikasi secara berkala untuk memastikan aplikasi tetap berjalan dengan baik.

1.7 Sistematika Penulisan

Pola penyajian penulisan laporan penelitian tugas akhir ini terbagi ke dalam sejumlah bab dan sub bab dengan susunan sebagai berikut:

a. Bab I Pendahuluan

   Bab ini berisi tentang gambaran umum tentang topik yang dibahas. Mencakup tentang pembahasan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan.

b. Bab II Landasan Teori

   Bab ini menguraikan teori-teori dasar yang relevan dengan penelitian ini, termasuk konsep dasar keamanan siber, Common Vulnerabilities and Exposures (CVE), teknologi AI generatif, serta studi-studi terkait yang menjadi dasar dalam pengembangan sistem.

c. Bab III Metodologi Penelitian

   Bab ini menjelaskan persyaratan dan perencanaan sistem, bersama dengan metodologi penelitian dan teknik pengembangan sistem. Ini mencakup analisis kebutuhan input, analisis kebutuhan output, analisis kebutuhan proses, dan desain pengembangan sistem. Tahapan-tahapan yang digunakan dalam metodologi *Waterfall* juga dijelaskan di sini.

d. Bab IV Hasil dan Pembahasan

   Bab ini menguraikan implementasi sistem yang telah dirancang, serta temuan dari penilaian sistem. Bab ini mencakup hasil pengujian fungsionalitas dan kinerja sistem, serta analisis dan diskusi mengenai efektivitas sistem dalam mengidentifikasi dan menganalisis kerentanan CVE.

e. Bab V Kesimpulan dan Saran

   Bab ini menyimpulkan temuan utama dari penelitian ini dan menjawab rumusan masalah yang telah diidentifikasi. Bagian ini juga memberikan rekomendasi untuk pengembangan sistem di masa depan agar sistem dapat terus ditingkatkan dan dioptimalkan dengan cara yang lebih baik.

BAB II
LANDASAN TEORI

2.1 Literatur Review

Dalam melakukan penelitian ini, penulis melakukan eksplorasi dan tinjauan literatur terkait beberapa penelitian yang memiliki topik pengembangan yang sama. Beberapa hasil dari eksplorasi tersebut disajikan di Tabel 2.1.

Tabel 2. 1 Tabel Literatur Review

No	Penulis	Judul Penelitian	Tahun	Metodologi	Hasil penelitian
1	Nan Tang, Chenyuan Yang, Ju Fan, Lei Cao	VeriFAI: Verified Generative AI	2023	Studi literatur dan eksperimen	Mengembangkan model AI generatif yang terverifikasi untuk berbagai aplikasi termasuk keamanan siber.
2	Boyang Chen, Zongxiao Wu, Ruoran Zhao	From Fiction to Fact: The Growing Role of Generative AI in Business	2023	Studi kasus dan analisis data	Menunjukkan bagaimana AI generatif digunakan dalam bisnis termasuk mitigasi risiko keamanan.
3	Pitikorn Khlaisamniang, Prachaya Khomduean, Kriangkrai Saetan, Supasin Wonglapsuwan	Generative AI for Self-Healing Systems	2023	Studi kasus dan eksperimen	Mengembangkan sistem self-healing untuk pertahanan siber otomatis menggunakan AI.
4	G.Harshvardhan, Mahendra Kumar Gourisaria, M. Pandey, S. Rautaray	A comprehensive survey and analysis of generative models in machine learning	2020	Studi literatur dan eksperimen	Mengkaji berbagai model pembelajaran mesin untuk deteksi dan mitigasi serangan siber.

No	Penulis	Judul Penelitian	Tahun	Metodologi	Hasil penelitian
5	Jandl, Alexander	Exploring Large Language Models for Inferring Relations Between Cybersecurity Constructs	2024	Eksperimen dan simulasi	Menggunakan model NLP seperti GPT untuk menganalisis deskripsi dalam catatan CVE guna meningkatkan proses identifikasi, analisis, dan mitigasi ancaman.
6	Kylie McClanahan, Sky Elder, Marie Louise Uwibambe, Yaling Liu, Rithyka Heng, Qinghua Li	When ChatGPT Meets Vulnerability Management: the Good, the Bad, and the Ugly	2024	Studi kasus dan analisis data	Studi tentang bagaimana ChatGPT digunakan untuk mengelola informasi CVE dan langkah-langkah mitigasi.
7	Yatin Bajaj, Manoj Samal	Accelerating Software Quality: Unleashing the Power of AI	2023	Studi kasus dan eksperimen	Mempercepat peningkatan kualitas perangkat lunak dengan memanfaatkan AI generatif.

Berdasarkan Tabel 2.1 dapat dilihat perbandingan penelitian ini dengan penelitian terdahulu. Pada penelitian terdahulu tidak terdapat fokus spesifik pada integrasi model AI generatif untuk analisis dan mitigasi CVE seperti dalam penelitian ini. Misalnya, penelitian oleh Nan Tang et al. (2023) mengembangkan model AI generatif yang terverifikasi untuk berbagai aplikasi termasuk keamanan siber, namun tidak secara khusus menargetkan analisis CVE. Penelitian oleh Boyang Chen et al. (2023) menunjukkan bagaimana AI generatif digunakan dalam bisnis termasuk mitigasi risiko keamanan, tetapi tidak mengeksplorasi secara mendalam bagaimana AI generatif dapat digunakan untuk mengelola data CVE.

Penelitian Pitikorn Khlaisamniang et al. (2023) mengembangkan sistem self-healing untuk pertahanan siber otomatis menggunakan AI, yang memberikan dasar yang kuat untuk aplikasi AI dalam keamanan siber, tetapi tidak secara langsung menangani pengelolaan CVE. G. Harshvardhan et al. (2020) mengkaji berbagai model pembelajaran mesin untuk deteksi dan

mitigasi serangan siber, yang relevan dengan penelitian ini, namun pendekatan mereka lebih umum dan tidak spesifik pada AI generatif atau analisis CVE.

Penelitian oleh Jandl, Alexander (2024) menggunakan model NLP seperti GPT untuk menganalisis deskripsi dalam catatan CVE guna meningkatkan proses identifikasi, analisis, dan mitigasi ancaman, yang lebih sejalan dengan penelitian ini. Namun, penelitian ini lebih fokus pada penggunaan NLP tanpa memperluas ke aplikasi generatif AI lainnya. Kylie McClanahan et al. (2024) mengeksplorasi penggunaan ChatGPT untuk mengelola informasi CVE dan langkah-langkah mitigasi, yang menunjukkan potensi besar tetapi tidak mengintegrasikan aspek verifikasi model AI. Terakhir, Yatin Bajaj dan Manoj Samal (2023) membahas percepatan peningkatan kualitas perangkat lunak dengan memanfaatkan AI generatif, namun tidak menekankan pada aspek keamanan siber atau manajemen CVE.

Penelitian ini memberikan kontribusi unik dengan mengintegrasikan AI generatif untuk analisis dan mitigasi CVE, menawarkan pendekatan yang lebih terfokus dan komprehensif untuk meningkatkan keamanan siber. Selebihnya, perbedaan fitur yang dimiliki antara penelitian ini dengan penelitian sebelumnya adalah pendekatan khusus untuk pengelolaan dan mitigasi CVE menggunakan AI generatif, yang belum banyak dieksplorasi dalam penelitian terdahulu.

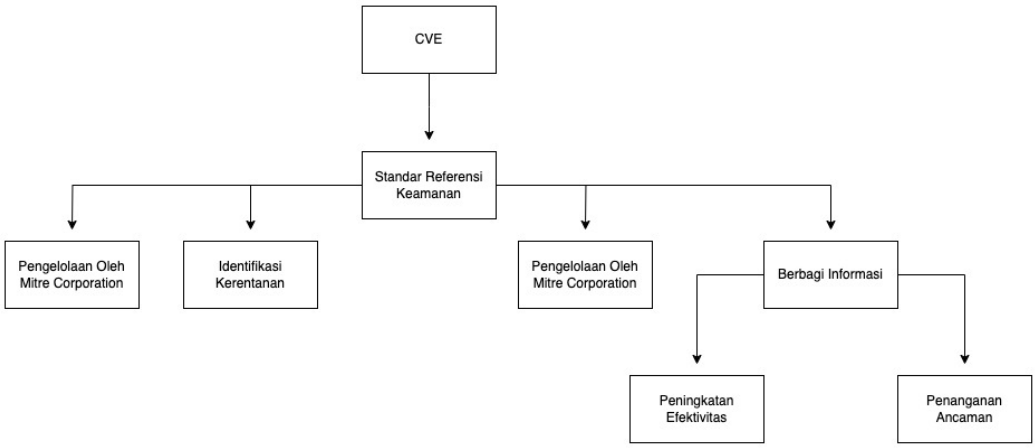
2.2 Keamanan Siber

Keamanan siber adalah praktik yang bertujuan untuk melindungi sistem, jaringan, dan program dari serangan digital. Serangan ini biasanya bertujuan untuk mengakses, mengubah, atau menghancurkan informasi sensitif, memeras uang dari pengguna, atau mengganggu proses bisnis yang normal. Dalam era digital yang semakin berkembang, keamanan siber menjadi sangat penting karena meningkatnya jumlah dan kompleksitas ancaman siber. Serangan siber tidak hanya semakin sering terjadi tetapi juga semakin canggih, dengan penyerang yang menggunakan teknologi terbaru untuk mengeksploitasi kerentanan dalam sistem. Menurut studi oleh (Couce-Vieira dkk., 2020), serangan siber dapat berdampak signifikan pada integritas, kerahasiaan, dan ketersediaan informasi, yang merupakan tiga pilar utama keamanan informasi. Integritas memastikan bahwa data tidak diubah atau dirusak oleh pihak yang tidak berwenang. Kerahasiaan melindungi informasi dari akses yang tidak sah, dan ketersediaan memastikan bahwa data dan sistem tetap dapat diakses oleh pengguna yang berhak saat dibutuhkan. Ketiga pilar ini sangat penting untuk menjaga kepercayaan pengguna dan menjaga kelangsungan operasi bisnis. Upaya untuk memperkuat keamanan siber harus dilakukan secara

terus-menerus dan adaptif terhadap perkembangan ancaman. Ini mencakup kemanan terbaru, pelatihan karyawan tentang praktik keamanan terbaik, dan monitoring secara terus-menerus untuk mendeteksi dan merespon ancaman. Teknologi AI generatif, dapat memainkan peran penting dalam analisis ancaman dan pengembangan strategi mitigasi yang lebih efektif, karena kemampuan AI untuk menganalisis data dalam skala besar dan mengidentifikasi pola yang mungkin terlewatkan oleh analisis manual. Selain itu, Integerasi AI dalam keamanan siber dapat membantu organisasi untuk tetap selangkah lebih maju dari penyerang dengan menyediakan wawasan yang lebih mendalam dan rekomendasi yang lebih cepat dan tepat waktu. Dengan demikian, keamanan siber yang efektif memerlukan pendekatan yang holistik dan dinamis, menggabungkan teknologi canggih dengan kebijakan yang ketat dan kesadaran keamanan yang tinggi di seluruh organisasi.

2.3 Common Vulnerabilities and Exposures (CVE)

Untuk mendukung upaya keamanan siber, salah satu mekanisme penting yang digunakan adalah Common Vulnerabilities and Exposure. Common Vulnerabilities and Exposures (CVE) adalah daftar standar dari kerentanan dan eksposur yang diketahui secara publik. CVE dikelola oleh MITRE Corporation dan digunakan sebagai referensi umum untuk meningkatkan keamanan siber. Setiap entri CVE mencakup deskripsi singkat dari kerentanan, dampaknya, dan referensi ke sumber daya lain yang terkait. CVE dikelola oleh Mitre Corporation (MITRE, n.d.) dan digunakan secara luas oleh industri keamanan siber untuk mengidentifikasi dan mengklasifikasikan kerentanan dalam perangkat lunak dan perangkat keras. Setiap entri CVE mendeskripsikan kerentanan spesifik yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab. Menurut (Lim dkk., 2023a), dengan adanya standar CVE, organisasi dapat lebih mudah dalam mengidentifikasi, mengelola, dan membagikan informasi mengenai kerentanan yang ditemukan, sehingga dapat meningkatkan efektivitas dalam menangani ancaman keamanan.



Gambar 2.1 Common Vulnerabilities and Exposures (CVE)

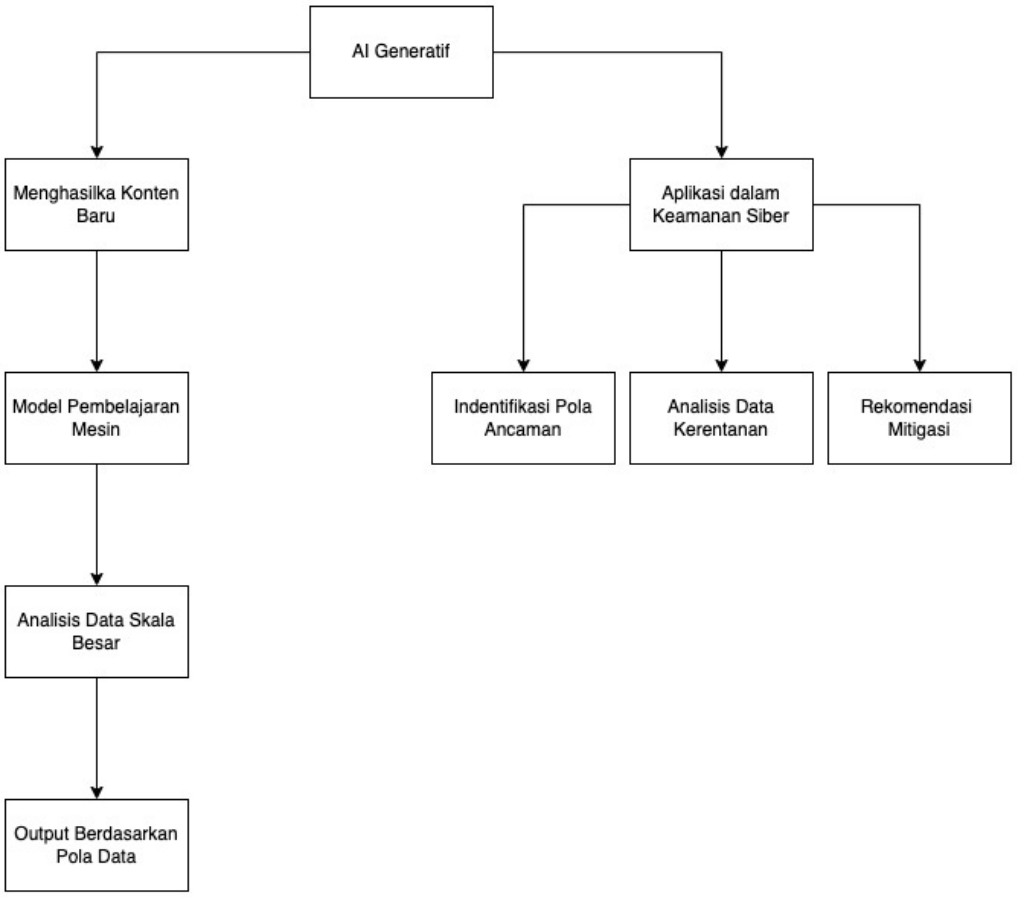
2.4 National Vulnerability Database (NVD)

Sebagai pendukung standar CVE, National Vulnerability Database (NVD) adalah repositori yang dikelola oleh National Institute of Standards and Technology (NIST) yang menyediakan standar informasi tentang kerentanan keamanan yang dilaporkan dalam produk perangkat lunak. NVD menggunakan data CVE dan memperkaya informasi tersebut dengan berbagai metrik, seperti Common Vulnerability Scoring System (CVSS) yang memberikan penilaian tingkat keparahan kerentanan. CVSS menyediakan skala standar untuk menilai dampak dan eksposur yang terkait dengan kerentanan tertentu, yang memudahkan organisasi untuk mengidentifikasi dan memprioritaskan upaya mitigasi berdasarkan tingkat risiko yang terukur. NVD juga menyediakan API yang memungkinkan akses otomatis ke data CVE untuk analisis dan pemantauan. Dengan API ini, pengembang dan peneliti keaamanan dapat mengintegerasikan data kerentanan langsung ke dalam alat analisis mereka, memungkinkan pemantauan terus-menerus dan respon cepat terhadap ancaman keamanan baru. Menurut Lim dkk. (2023) (Lim dkk., 2023b), API NVD memfasilitasi akses informasi kerentanan yang terstruktur dan terstandarisasi, sehingga memudahkan. Organisasi dalam melakukan analisis risiko dan implementasi langkah-langkah mitigasi yang sesuai. Selain itu, NVD memainkan peran penting dalam ekosistem keaamanan siber dengan menyediakan sumber data yang dapat diandalkan untuk komunitas global. Data yang disediakan oleh NVD sering digunakan dalam penelitian keamanan, pengembangan alat deteksi ancaman, dan berbagai solusi keamanan komersial yang dirancang untuk melindungi sistem informasi dari eksploitasi. Menurut National Institute of Standards and Technology (NIST), keberadaan NVD meningkatkan kemampuan untuk berbagi informasi tentang ancaman keaamanan secara lebih luas dan efektif, yang merupakan komponen penting dalam upaya kolektif untuk meningkatkan postur

keamanan siber global. Dengan adanya NVD, data CVE dapat diakses secara lebih terstruktur dan terstandarisasi, yang sangat penting dalam upaya untuk meningkatkan keaamnanan perangkat lunak dan sistem informasi di berbagai organisasi.

2.5 *Artificial Intelligence* Generatif

Seiring dengan data yang terstruktur dan terstandarisasi yang disediakan oleh NVD, pemanfaatan teknologi *Artificial Intelligence* (AI) generatif menjadi semakin relevan dalam analisis data keamanan. AI generatif adalah teknologi AI yang mampu menghasilkan konten baru berdasarkan data yang ada. Teknologi ini menggunakan model pembelajaran mesin untuk memproses dan menganalisis data dalam skala besar, kemudian menghasilkan output yang sesuai dengan pola yang ditemukan dalam data. Menurut (Gupta dkk., 2023a), AI generatif dapat digunakan dalam berbagai aplikasi, termasuk keamanan siber, untuk mengidentifikasi pola-pola tertentu yang mungkin terlewatkan oleh analisis manual dalam konteks keamanan siber, AI generatif dapat sangat penting karena memiliki kemampuan untuk meningkatkan deteksi respon terhadap ancaman siber. Dengan menggunakan model prediktif yang canggih, AI generatif dapat menganalisis data kerentanan dan menghasilkan rekomendasi mitigasi yang lebih akurat dan efisien (Gupta dkk., 2023b). AI dan alat otomatisasi keamanan telah terbukti mempercepat deteksi dan penanganan pelanggaran, mengurangi siklus hidup pelanggaran secara signifikan, yang sangat penting dalam konteks ancaman seperti pelanggaran rantai pasokan perangkat lunak yang bisa berdampak besar pada keuangan dan reputasi organisasi (Oh & Shon, 2023). Selain itu AI generatif juga dapat mengantisipasi serangan yang lebih kompleks melalui teknik seperti pemodelan prediktif dan analisis pola. Teknologi ini tidak hanya dapat mendeteksi ancaman yang sudah dikenal, tetapi juga memprediksi ancaman baru yang belum pernah terlihat sebelumnya, memberikan tim keamanan konteks yang diperlukan untuk bertindak segera (Dunmore dkk., 2023). Contoh konkret dari hal ini adalah penggunaan AI generatif dalam mengembangkan *malware polymorphic* yang dapat berubah-ubah, menunjukan betapa kuatnya teknologi ini jika disalahgunakan, sekaligus menyoroti perlunya perlunya pemanfaatan AI generatif yang bijak dan bertanggung jawab dalam keamanan siber. Dengan demikian, penerapan AI generatif dalam keamanan siber bukan hanya tentang peningkatan efisiensi dan akurasi, tetapi juga tentang menciptakan lapisan pertahanan tambahan yang mampu menghadapi ancaman yang semakin kompleks dan canggih. Hal ini menjadikan AI generatif sebagai komponen vital dalam strategi keamanan siber modern.



Gambar 2.2 Artificial Intelligence Generatif

2.6 Model GPT (Generative Pre-trained Transformer)

Salah satu implemnetasi AI generatif adalah model *Generative Pre-trained Transformer* (GPT) yang dikembangkan oleh OpenAI. Model GPT menggunakan arsitektur Transformer, yang pertama kali diperkenalkan oleh, untuk memungkinkan pehaman konteks dan makna teks secara mendalam melalui mekanisme *self-attention*. Arsitektur ini sangat efektif dalam menangkap hubungan antar kata dalam teks, baik dalam jangka pendek maupun panjang.

2.6.1 Pelatihan dan Adaptasi

Model GPT dilatih menggunakan Teknik pembelajaran tanpa pengawasan pada kumpulan data teks yang sangat besar, seperti *Common Crawl*, yang terdiri dari milyaran kata dari berbagai domain. Proses pelatihan ini memungkinkan model untuk membangun representasi yang kaya dari Bahasa alami, menangkap nuansa dan variasi dalam penggunaan Bahasa. Setelah tahap pelatihan dasar, model GPT dapat disesuaikan atau dikenal dengan *fine-tuned* dengan tugas spesifik seperti penjawaban pertanyaan, terjemahan, dan generasi teks, melalui pembelajaran terawasi dengan dataset yang lebih kecil dan terfokus (Brown dkk., 2020).

2.6.2 Aplikasi dalam Keamanan Siber

Dalam penelitian ini, model GPT digunakan untuk menghasilkan analisis dan rekomendasi berdasarkan data CVE (Common Vulnerabilities and Exposures). Pengguna GPT dalam konteks ini menawarkan beberapa keuntungan utama:

a. Analisis Cepat dan Akurat:

   GPT dapat menganalisis deskripsi kerentanan dan menghasilkan ringkasan serta rekomendasi mitigasi yang akurat dan relevan. Ini membantu tim keamanan untuk memahami ancaman dan tindakan yang perlu diambil dengan lebih cepat. Dengan kemampuan untuk menganalisis sejumlah besar data dalam waktu singkat, GPT memfasilitasi identifikasi kerentanan yang lebih cepat dan akurat dibandingkan dengan metode manual tradisional.

b. Pemahaman Kontekstual:

   Dengan kemampuannya memahami konteks dan makna teks, GPT dapat mengidentifikasi hubungan antara berbagai kerentanan dan mengevaluasi dampaknya terhadap sistem secara keseluruhan. Ini penting untuk mengembangkan strategi mitigasi yang komprehensif dan efektif. Pemahaman kontekstual ini memungkinkan GPT untuk menilai ancaman dengan lebih baik dan memberikan rekomendasi yang lebih tepat sasaran.

c. Otomatisasi Proses Keamanan:

   Model GPT dapat mengotomatisasi tugas-tugas yang biasanya memerlukan intervensi manusia, seperti penulisan laporan keamanan dan penilaian risiko. Otomatisasi ini meningkatkan efisiensi dan memungkinkan tim keamanan fokus pada tugas-tugas strategis lainnya. Dengan mengurangi beban kerja manual, GPT memungkinkan respons yang lebih cepat terhadap ancaman keamanan yang muncul.

Sebagai studi kasus, model GPT telah digunakan dalam berbagai proyek penelitian dan aplikasi industri untuk mengatasi tantangan keamanan siber. Sebagai contoh, penelitian oleh Radford dkk. menunjukkan bahwa model GPT dapat digunakan untuk deteksi anomali dalam jaringan sistem IT, serta dalam simulasi serangan siber untuk menguji ketahanan sistem. Implementasi model GPT dalam konteks CVE di penelitian ini diharapkan dapat memberikan wawasan baru dan solusi yang lebih efektif dalam pengelolaan kerentanan siber.

Dengan demikian, model GPT menawarkan kontribusi yang signifikan dalam memperkuat strategi keamanan siber melalui analisis yang lebih canggih dan otomatisasi yang lebih luas. Hal ini menjadikannya alat yang sangat berguna dalam mitigasi risiko dan peningkatan postur

keamanan organisasi. Penggunaan GPT untuk analisis data CVE membuka peluang baru dalam identifikasi kerentanan yang lebih cepat dan efektif, serta dalam pengembangan solusi keamanan yang lebih responsif dan proaktif.

2.7 OpenAI

Dalam konteks ini OpenAI API menjadi sangat relevan. OpenAI API adalah layanan yang menyediakan akses ke model pembelajaran mesin canggih yang dikembangkan oleh OpenAI. API ini memungkinkan pengembang untuk memanfaatkan kemampuan model seperti GPT untuk berbagai aplikasi, termasuk analisis teks, pemrosesan bahasa alami, dan penemuan kerentanan. Model GPT-4, misalnya, memiliki kemampuan untuk memahami dan menghasilkan teks dengan konteks yang kompleks. Salah satu produk utamanya adalah OpenAI API, yang memberikan akses ke model bahasa AI canggih seperti GPT-3 dan GPT-4. Model-model ini mampu menghasilkan teks yang koheren dan relevan berdasarkan input yang diberikan, sehingga sangat berguna dalam berbagai aplikasi, seperti analisis teks, pembuatan konten, dan asistensi interaktif. Dalam penelitian ini, OpenAI API digunakan untuk menganalisis data CVE dan memberikan rekomendasi mitigasi berdasarkan deskripsi kerentanan (Gupta dkk., 2023). Penggunaan OpenAI API memperkuat peran model GPT dalam keamanan siber, menjadikannya alat yang tidak hanya mampu menganalisis dan mengidentifikasi kerentanan dengan lebih efektif, tetapi juga menyediakan solusi yang lebih tepat dan proaktif. Integrasi antara analisis canggih oleh GPT dan aksesibilitas melalui OpenAI API menciptakan sinergi yang meningkatkan kemampuan organisasi untuk mengelola dan mengurangi risiko keamanan dengan lebih efisien dan responsif.

2.8 Implementasi Sistem dengan Flask dan OpenAI

Dalam penelitian ini, pengembangan sistem dilakukan dengan menggunakan kerangka kerja Flask dan OpenAI API. Berikut adalah beberapa komponen kunci dari sistem yang dikembangkan:

2.8.1 Flask sebagai Kerangka Kerja Web

Flask adalah kerangka kerja web berbasis Python yang ringan dan fleksibel, digunakan untuk mengembangkan aplikasi web yang dapat menangani permintaan HTTP, routing, dan rendering template. Dalam konteks penelitian ini, Flask digunakan untuk membangun aplikasi yang menerima input pengguna, memproses data CVE, dan mengembalikan hasil analisis

kepada pengguna. Flask memungkinkan pengembangan yang cepat dan mudah berkat sifatnya yang minimalis dan modular, sehingga memudahkan penambahan fungsionalitas sesuai kebutuhan penelitian (Chauhan dkk., 2019)(Sholeh & Suraya, n.d.).

2.8.2 Integrasi OpenAI API

OpenAI API digunakan untuk memanfaatkan model AI generatif dalam menganalisis data CVE. Model AI ini dapat memahami deskripsi kerentanan dan memberikan rekomendasi mitigasi yang lebih tepat. Integrasi OpenAI API dalam sistem memungkinkan aplikasi memproses permintaan secara dinamis dan memberikan respons yang relevan berdasarkan data CVE yang dianalisis (Gupta dkk., 2023). OpenAI API menawarkan kemampuan untuk menghasilkan teks yang koheren dan informatif, yang sangat berguna dalam memberikan penjelasan mendetail mengenai kerentanan dan langkah mitigasinya.

2.8.3 Pengumpulan dan Pemrosesan Data CVE

Data CVE dikumpulkan dari National Vulnerability Database (NVD) menggunakan API yang tersedia. Data ini mencakup informasi tentang kerentanan, seperti deskripsi, tingkat keparahan, dan tanggal publikasi. Proses pemrosesan data melibatkan beberapa langkah, termasuk pengambilan data, ekstraksi detail, dan analisis menggunakan model AI generatif. Berikut adalah detail dari masing-masing langkah tersebut:

a. Pengambilan Data

Dengan menggunakan API dari NVD, sistem mengumpulkan data CVE yang relevan berdasarkan permintaan pengguna atau *file* CSV yang diunggah. Data yang diambil mencakup ID CVE, deskripsi, tanggal publikasi, tingkat keparahan, dan referensi terkait.

b. Ekstraksi Detail

Setelah data diambil, sistem mengekstraksi informasi penting dari setiap entri CVE. Informasi ini mencakup ID CVE, deskripsi kerentanan, tingkat keparahan berdasarkan skala CVSS (Common Vulnerability Scoring System), dan vektor serangan. Ekstraksi dilakukan menggunakan fungsi khusus yang dirancang untuk menguraikan struktur data JSON yang dikembalikan oleh API NVD (Sumoto dkk., 2022).

c. Analisis Menggunakan AI Generatif

Data yang telah diekstraksi dianalisis menggunakan model AI generatif melalui OpenAI API. Model ini memberikan ringkasan dan rekomendasi mitigasi berdasarkan deskripsi dan detail kerentanan. Rekomendasi tersebut mencakup langkah-langkah spesifik yang bisa

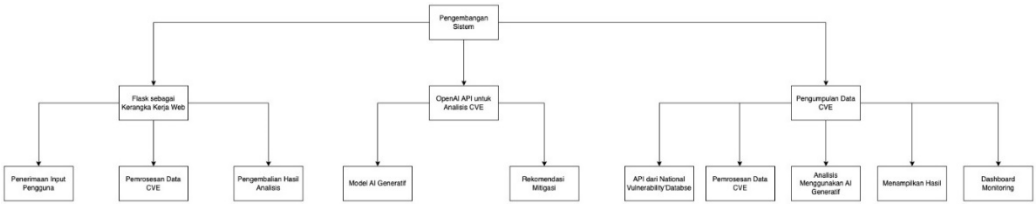
diambil untuk mengurangi risiko yang terkait dengan kerentanan, seperti pembaruan perangkat lunak, konfigurasi ulang sistem, atau penerapan patch keamanan (McKee & Noever, 2023).

d. Presentasi Hasil

Hasil analisis disajikan kepada pengguna melalui antarmuka web yang dibangun dengan Flask. Pengguna dapat melihat ringkasan kerentanan, rekomendasi mitigasi, serta referensi tambahan untuk informasi lebih lanjut. Sistem ini juga memiliki fitur untuk mengunggah file CSV yang berisi daftar CVE, yang kemudian diproses dan dianalisis secara batch.

e. Dasbor Monitoring

Sistem ini juga mencakup Dasbor monitoring untuk melihat pembaruan terbaru dalam database CVE. Pengguna dapat memantau kerentanan yang baru diterbitkan dan mendapatkan informasi tentang kerentanan yang relevan dengan sistem mereka (Ablahd & Dawwod, 2020). Dengan menggabungkan Flask dan OpenAI API, penelitian ini menghasilkan sistem yang mampu memberikan analisis dan rekomendasi mitigasi CVE secara efisien dan akurat. Integrasi teknologi ini memungkinkan organisasi untuk lebih proaktif dalam mengidentifikasi dan mengatasi kerentanan keamanan, sehingga meningkatkan keamanan siber secara keseluruhan.



Gambar 2.3 Implementasi Sistem dengan Flask dan OpenAI

2.9 Python

Python adalah Bahasa pemrograman tingkat tinggi yang sangat popular dan serbaguna, digunakan secara luas dalam berbagai bidang seperti pengembangan aplikasi web, analisis data, kecerdasan buatan, dan otomatisasi tugas. Kelebihan python yang membuatnya populer antara lain adalah sintaksnya yang sederhana dan mudah dibaca serta ekosistem yang sangat luas dan kaya (Peta, 2022). Python dikenal dengan sintaks yang intuitif dan mirip dengan Bahasa inggris, sehingga memudahkan pemula untuk belajar dan menguasai Bahasa ini dengan cepat. Keterbacaan kode python memungkinkan untuk menulis kode yang jelas dan mudah dipahami, yang juga membantu dalam proses pemeliharaan kode.

2.9.1 Dukungan Pustaka yang Luas

Python memiliki dukungan pustaka yang sangat luas, yang mencakup berbagai kebutuhan pengembangan. Beberapa Pustaka penting yang digunakan dalam penelitian ini adalah:

a. Flask: Sebuah Pustaka micro framework untuk pengembangan aplikasi web. Flask memfasilitasi penanganan permintaan HTTP, routing, dan rendering template, memungkinkan pengembang untuk membangun aplikasi web yang ringan namun kuat.
b. Pandas: Pustaka yang digunakan untuk manipulasi data. Pandas menyediakan struktur data dan fungsi yang kuat untuk mengolah data dalam berbagai format, sehingga sangat berguna dalam analisis data. Dalam konteks ini, Pandas digunakan untuk membaca dan memproses data dari file CSV, yang memudahkan analisis dan manipulasi data, seperti filtering atau transformasi data sebelum dimasukan ke dalam database (Lau dkk., 2023).
c. OpenAI API: Pustaka yang memungkinkan interaksi dengan model-model AI generatif dari OpenAI, seperti GPT, untuk analisis dan generasi teks berdasarkan data yang diberikan.

Pada penelitian ini, Python digunakan sebagai Bahasa utama untuk mengembangkan aplikasi web menggunakan Flask, serta untuk memproses data CVE (Common Vulnerabilities and Exposures) dan berinteraksi dengan OpenAI API. Berikut adalah beberapa peran penting Python dalam penelitian ini:

a. Pengembangan Aplikasi Web: Menggunakan Flask, Python memfasilitasi pembangunan aplikasi web yang dapat menangani permintaan HTTP, melakukan routing, dan merender template. Flask juga memungkinkan integrasi yang mudah dengan Pustaka lain untuk memperkaya fungsi aplikasi.
b. Pemrosesan Data CVE: Dengan menggunakan Pandas dan Pustaka data lainnya, Python memungkinkan manipulasi dan analisis data CVE dengan efisien. Hal ini penting untuk mengidentifikasi pola dan hubungan dalam data yang dapat digunakan untuk meningkatkan keamanan sistem.
c. Interaksi dengan OpenAI API: Python memfasilitasi interaksi dengan OpenAI API untuk menggunakan model GPT dalam menganalisis dan memberikan rekomendasi berdasarkan data CVE. Kemampuan ini membantu dalam otomatisasi analisis keamanan dan pengembangan strategi mitigasi yang lebih efektif.

2.10 Werkzeug Security

Werkzeug adalah toolkit WSGI untuk Python yang menyediakan berbagai modul keamanan, termasuk fungsi hashing dan verifikasi kata sandi. Dalam konteks aplikasi ini, ‘generate_password_hash’ digunakan untuk mengamankan kata sandi pengguna dengan meng-hash-nya sebelum disimpan ke database, sementara ‘check_password_hash’ digunakan untuk memverifikasi kata sandi yang dimasukkan oleh pengguna dengan hash yang tersimpan. Metode hashing yang digunakan adalah PBKDF2, yang diakui secara luas dan direkomendasikan untuk keamanan kata sandi. Menurut Rooparaghunath dkk (2023) (Rooparaghunath dkk., 2023), PBKDF2 adalah salah satu algoritma hashing yang paling aman dan efektif untuk melindungi kata sandi, karena menggunakan salt dan iterasi untuk meningkatkan keamanan hash.

2.11 SERP API

SERP API adalah alat yang digunakan untuk mengakses hasil pencarian web dari berbagai mesin pencari melalui antarmuka pemrograman aplikasi. Ini memungkinkan aplikasi untuk mengirim kueri pencarian dan menerima data terstruktur yang berisi hasil pencarian, yang dapat digunakan untuk analisis lebih lanjut atau ditampilkan kepada pengguna. Dalam penelitian ini, SERP API digunakan untuk analisis lebih lanjut atau ditampilkan kepada pengguna. Dalam penelitian ini, SERP API digunakan untuk memperoleh informasi terkait kerentanan yang ditemukan di CVE dari berbagai sumber di internet, memungkinkan analisis yang lebih komprehensif dan terkini. Oliviera & Teixeira Lopes (2023) (Oliveira & Teixeira Lopes, 2023) menjelaskan bahwa penggunaan SERP API dapat meningkatkan efisiensi dan akurasi dalam pengumpulan data dari berbagai sumber web.

2.12 SQLAlchemy

SQLAlchemy adalah toolkit SQL dan Object-Relational Mapping (ORM) untuk bahasa pemrograman Python. ORM memungkinkan pengembang untuk berinteraksi dengan database menggunakan objek Python, alih-alih menulis SQL secara langsung. SQLAlchemy menyediakan API yang kuat dan fleksibel untuk definisi model, eksekusi query, dan manajemen hubungan antar tabel. Dalam penelitian ini, SQLAlchemy digunakan untuk mengelola data pengguna, thread, dan CVE dalam aplikasi. Penggunaan ORM seperti SQLAlchemy mempermudah pengelolaan data dalam aplikasi dan meningkatkan produktivitas pengembang dengan menyediakan abstraksi yang lebih tinggi dari interaksi database.

Dolhopolov & Imanhulova (Dolhopolov & Imanhulova, 2023) menyatakan bahwa SQLAlchemy adalah alat yang sangat efisien untuk pengembangan aplikasi berbasis database dengan Python, karena kemampuannya untuk mengelola skema database yang kompleks dengan cara yang intuitif

2.13 *AI Prompting*

AI prompting adalah teknik untuk memberikan instruksi atau pertanyaan yang jelas dan spesifik kepada model AI untuk memperoleh respons yang diinginkan. Teknik ini penting untuk memastikan keluaran yang dihasilkan oleh model AI relevan dan bermanfaat. Dalam konteks penelitian ini, prompting digunakan untuk meminta model OpenAI menghasilkan ringkasan dan rekomendasi mitigasi berdasarkan deskripsi kerentanan CVE. *Prompt* yang baik harus mencakup konteks yang cukup, instruksi yang jelas, dan contoh yang relevan jika diperlukan, untuk memandu model dalam menghasilkan respons yang berkualitas tinggi. Liu dkk. (2023) (Liu dkk., 2023) menekankan bahwa keberhasilan AI prompting sangat bergantung pada kualitas kejelasan prompt yang diberikan, yang dapat secara signifikan mempengaruhi kualitas output yang dihasilkan oleh model AI. Untuk mendapatkan hasil analisis yang maksimal dari model GPT seperti OpenAI, penting untuk memahami dan menggunakan teknik prompting yang tepat. Berikut adalah beberapa teknik utama yang digunakan dalam prompting:

a. *Zero-Shot Prompting*: Dalam Teknik ini model menerima deskripsi tugas tanpa contoh input-output. Model memanfaatkan pengetahuannya yang sudah ada untuk menghasilkan prediksi berdasarkan prompt yang diberikan. Teknik ini berguna untuk tugas-tugas baru yang tidak memiliki data pelatihan yang luas (Sahoo dkk., 2024).
b. *Few-Shot Prompting*: Teknik ini memberikan beberapa contoh input-output kepada model untuk membantu memahami tugas yang diberikan. Meskipun tidak seefisien pelatihan penuh, beberapa contoh berkualitas tinggi dapat meningkatkan kinerja model pada tugas-tugas kompleks. Namun, Teknik ini membutuhkan token tambahan untuk menyertakan contoh-contoh tersebut, yang bisa menjadi masalah untuk input teks yang Panjang (Sahoo dkk., 2024).
c. *Chain-of-Thought (CoT) Prompting*: CoT prompting membantu model dalam memproses penalaran yang kompleks dengan meminta model untuk memecah masalah menjadi langkah-langkah logis yang lebih kecil. Teknik ini telah terbukti efektif dalam meningkatkan pemahaman dan keluaran model dalam tugas-tugas yang membutuhkan penalaran berjenjang, seperti soal matematika yang kompleks (Sahoo dkk., 2024).

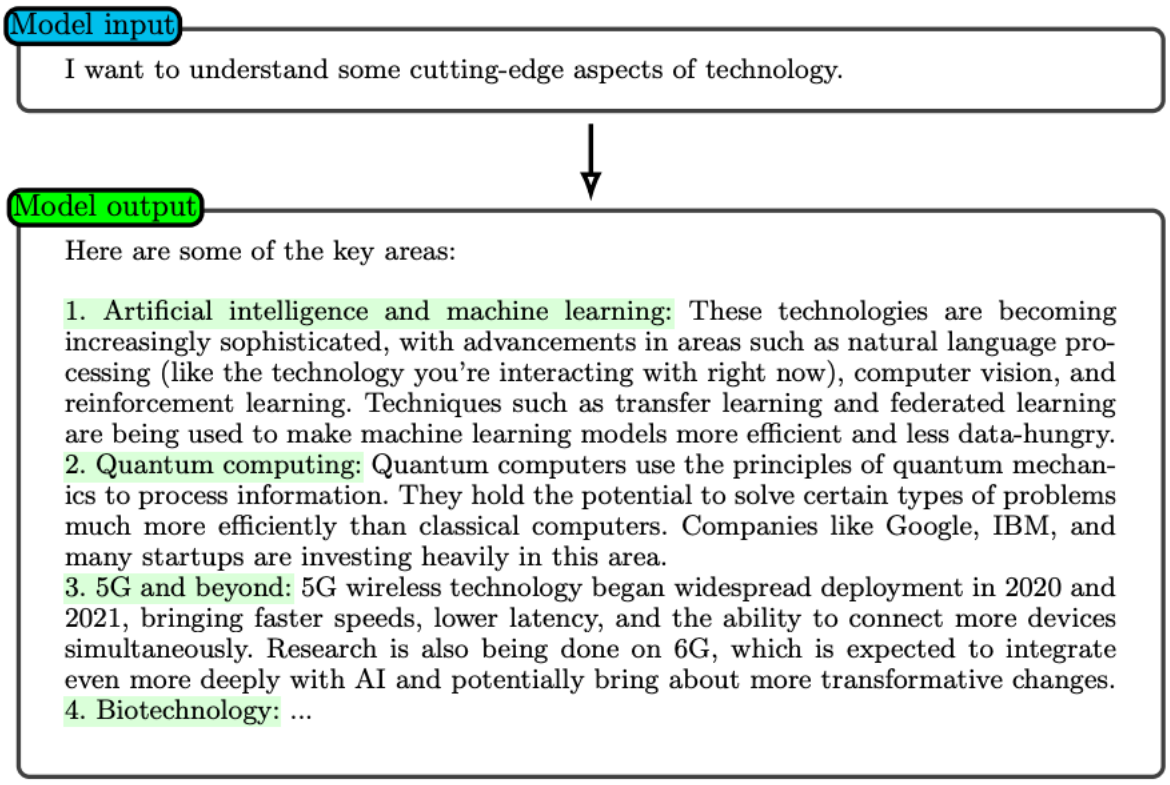
d. *Automatic Chain-of-Thought (Auto-CoT) Prompting*: Auto-CoT adalah pengembagan dari CoT yang mengotomatiskan proses pembuatan rantai penalaran. Teknik ini menggunakan prompt seperti "Let's think step-by-step" untuk memandu model dalam menghasilkan rantai penalaran. Auto-CoT meningkatkan robustitas dengan melakukan sampling yang beragam, sehingga mengurangi kesalahan dalam rantai penalaran yang dihasilkan (Sahoo dkk., 2024).
e. *Self-Consistency*: Teknik ini meningkatkan kinerja penalaran melalui sampling dari decoder model. Kemudian, Teknik ini mengidentifikasikan jawaban akhir yang paling konsisten dengan merata-rata rantai penalaran yang dihasilkan. Pendekatan ini telah terbukti meningkatkan akurasi secara signifikan pada berbagai macam benchmark penalaran yang kompleks (Sahoo dkk., 2024)

2.14 *Prompt Engineering*

Prompt engineering merupakan teknik yang sangat penting dalam pengembangan dan penggunaan model bahasa besar (LLMs) seperti GPT-4. Dengan hanya mengintegrasikan beberapa elemen kunci, seseorang dapat merancang prompt dasar yang memungkinkan LLMs menghasilkan jawaban berkualitas tinggi. Dalam konteks skripsi ini, *prompt engineering* digunakan untuk memaksimalkan efisiensi dan akurasi respons yang dihasilkan oleh model dalam analisis CVE.

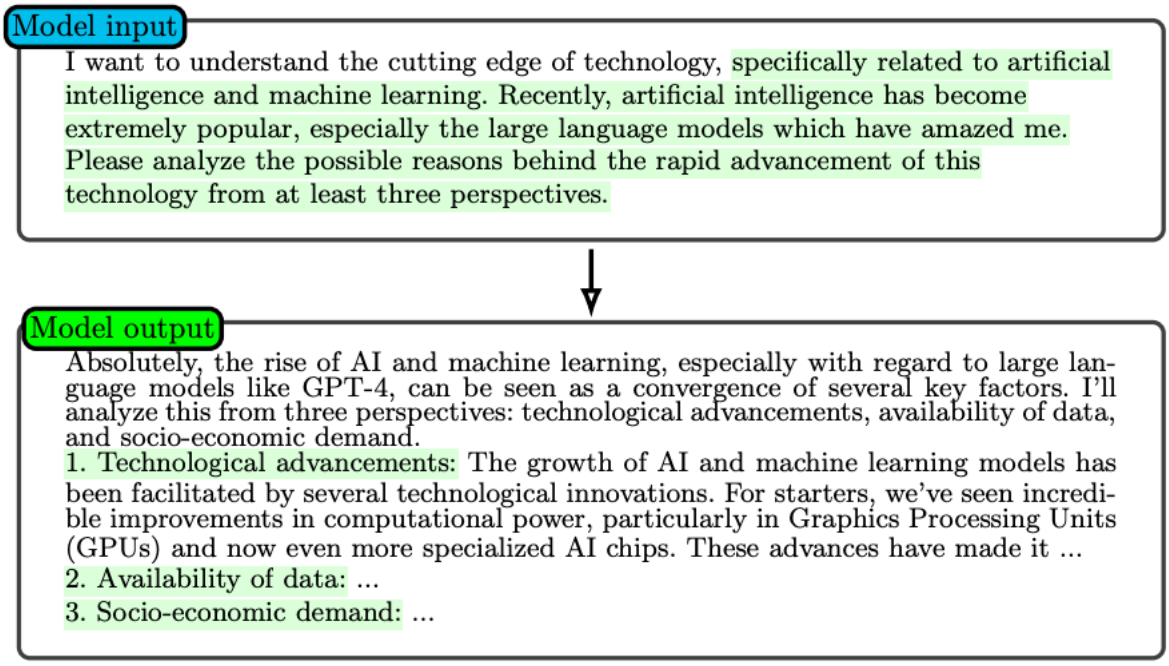
a. *Model Introduction GPT- 4*: Ketika GPT-4 menerima input *prompt*, teks input akan terlebih dahulu diubah menjadi token yang dapat diinterpretasikan dan di proses oleh model. Token ini kemudian dikelola oleh lapisan *transformer* yang menangkap hubungan dan konteks mereka. Di dalam lapisan ini, mekanisme perhatian mendistribusikan bobot yang berbeda ke token berdasarkan relevansi dan konteksnya. Setelah pemrosesan perhatian, model membentuk representasi dari data input, yang kemudian di-decode Kembali menjadi teks yang dapat dibaca manusia (Chen et al., 2023).
b. *Gving Instructions*: Metode memberikan instruksi, juga dikenal sebagai *re-reading*, mengacu pada heuristic strategi membaca manusia. Telah diamati bahwa *output* yang dihasilkan oleh GPT-4 cenderung terlalu umum ketika diberikan instruksi dasar tanpa deskripsi tambahan. Oleh karena itu, deskripsi yang komprehensif sangat penting untuk mendapatkan output yang lebih tepat dan relevan (Chen et al., 2023).
c. *Be Clear and Precise*: Metode dasar kedua dalam prompt engineering adalah "menjadi jelas dan tepat". Ini melibatkan perumusan prompt yang tidak ambigu dan spesifik, yang

dapat memandu model untuk menghasilkan konten yang lebih selaras dengan kebutuhan spesifik skenario tertentu, karena mengurangi ketidakpastian model dan mengarahkannya ke respons yang benar (Chen et al., 2023).



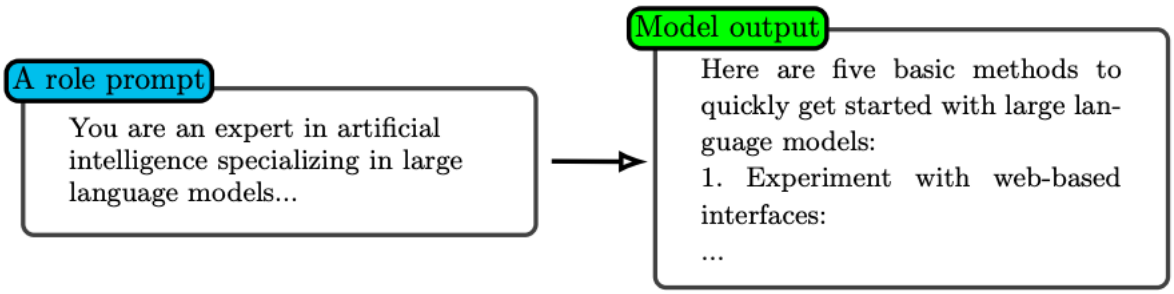
Gambar 2. 4 Memberikan Instruksi Tanpa Deskripsi Tambahan
Source: (Chen et al., 2023)

Skenario yang diberikan ini mengurangi ketidakpastian model dan mengarahkannya pada respons yang tepat. Sebagai contoh, seperti yang ditunjukkan pada Gambar 2, alih-alih mengajukan permintaan yang tidak jelas seperti "Saya ingin memahami teknologi terkini" pada Gambar 1, prompt yang lebih tepat akan berbunyi "Saya ingin memahami teknologi terkini, khususnya yang berkaitan dengan kecerdasan buatan dan pembelajaran mesin...".



Gambar 2. 5 Memberikan Instruksi Dengan Instruksi Tambahan
Source: (Chen et al., 2023)

d. *Role-Prompting*: *Role-prompting* melibatkan memberikan model peran spesifik untuk dimainkan, seperti asisten yang membantu atau pakar yang berpengetahuan luas. Metode ini efektif dalam membimbing respons model dan memastikan bahwa mereka sesuai dengan output yang diinginkan. Misalnya, jika model diminta bertindak sebagai sejarawan, kemungkinan besar model akan memberikan respons yang lebih rinci dan akurat secara kontekstual ketika ditanya tentang peristiwa sejarah (Chen et al., 2023).



Gambar 2. 6 *Role Prompting Source*: (Chen et al., 2023)

e. *Use of Triple Quotes to Separate*: Penggunaan *triple quotes* adalah teknik dalam *prompt engineering* yang digunakan untuk memisahkan bagian-bagian berbeda dari prompt atau untuk mengenkapsulasi string multi-baris. Teknik ini sangat berguna ketika menangani prompt yang kompleks yang mencakup beberapa komponen atau ketika prompt itu sendiri berisi kutipan(Chen et al., 2023).

f. *Try Several Times*: Karena sifat non-deterministik LLMs, sering kali bermanfaat untuk mencoba beberapa kali saat menghasilkan respons. Teknik ini, yang dikenal sebagai resampling, melibatkan menjalankan model beberapa kali dengan prompt yang sama dan memilih output terbaik. Pendekatan ini dapat membantu mengatasi variabilitas bawaan dalam respons model dan meningkatkan peluang memperoleh output berkualitas tinggi (Chen et al., 2023).
g. *One-Shot or Few-Shot Prompting*: *One-shot dan few-shot prompting* adalah dua teknik penting dalam prompt engineering. *One-shot prompting* mengacu pada metode di mana model diberikan satu contoh untuk dipelajari, sedangkan *few-shot prompting* memberikan model beberapa contoh. Pilihan antara one-shot dan *few-shot prompting* sering bergantung pada kompleksitas tugas dan kemampuan model. Untuk tugas yang sederhana atau model yang sangat mampu, one-shot prompting mungkin cukup. Namun, untuk tugas yang lebih kompleks atau model yang kurang mampu, *few-shot prompting* dapat memberikan konteks dan panduan tambahan, sehingga meningkatkan kinerja model (Chen et al., 2023).

Pemilihan prompt dalam skripsi ini didasarkan pada kebutuhan untuk mendapatkan jawaban yang tepat dan komprehensif dari model. Misalnya, prompt yang dirancang untuk mendapatkan analisis CVE mencakup elemen-elemen spesifik yang memastikan model dapat memahami konteks dan memberikan informasi yang akurat. Prompt seperti "Jelaskan kerentanan CVE-2023-1234" dirancang untuk memandu model dalam memberikan penjelasan terperinci mengenai suatu CVE tertentu, termasuk dampaknya, vektor serangan, dan langkah-langkah mitigasi yang direkomendasikan. Dengan menerapkan elemen-elemen kunci dalam prompt engineering, model dapat menghasilkan respons yang lebih relevan dan berguna untuk analisis CVE, sebagaimana didukung oleh penelitian Liu et al. (2023) yang menunjukkan bahwa prompt engineering dapat meningkatkan performa model bahasa dalam berbagai tugas.

BAB III
METODOLOGI PENELITIAN

Pada bagian ini akan diuraikan metodologi yang dilakukan pada pengembangan sistem yang mengintegrasi kecerdasan buatan generatif untuk analisis dan mitigasi data CVE. Untuk pengembangan ini, penulis menggunakan metode *waterfall* yang terdiri dari lima tahapan, yaitu *requirement* (analisis kebutuhan), desain, *implementation* (implementasi sistem), *testing* (pengujian), dan *maintenance* (pemeliharaan). Namun, penelitian ini hanya sampai pada tahap *testing* (pengujian) dan tidak memerlukan tahap *maintenance* (pemeliharaan). Hal ini disebabkan karena penelitian ini berfokus pada pengembangan sistem sampai pada tahap pengujian untuk mengetahui apakah sistem yang dikembangkan sudah memenuhi kebutuhan yang ditetapkan atau belum.

3.1 Metode Pembangunan Sistem

Metode *Waterfall*, yang dikenal sebagai "Model Sequential Linear" atau "siklus hidup klasik," pertama kali diperkenalkan sekitar tahun 1970. Meskipun dianggap sebagai model kuno, model ini tetap menjadi salah satu model yang paling sering digunakan dalam pengembangan perangkat lunak (Mudassar & Khan, 2023). Model *Waterfall* bersifat sistematis dan berurutan, dengan tahapan-tahapan yang dimulai dari persyaratan sistem, kemudian berlanjut ke analisis persyaratan, desain arsitektur sistem, implementasi, integrasi, dan pengujian sistem, serta evaluasi sistem sebagai langkah terakhir. Setiap tahapan harus diselesaikan sebelum tahapan berikutnya dimulai, dan tidak ada tahapan yang dilewati. Dalam model ini, fase proyek yang baru tidak akan dimulai sebelum fase sebelumnya diselesaikan secara penuh. Dengan pendekatan yang sistematis dan terstruktur, metode *Waterfall* memungkinkan pengembangan perangkat lunak yang lebih terorganisir dan terdokumentasi dengan baik, meskipun kurang fleksibel dalam menghadapi perubahan kebutuhan selama proses pengembangan.

3.2 Analisis Kebutuhan Sistem

Sistem yang dirancang adalah aplikasi berbasis web untuk analisis CVE (Common Vulnerabilities and Exposures) yang dilengkapi dengan fitur pencarian web, upload CSV, dan chat interaktif menggunakan OpenAI API. Aplikasi ini dikembangkan dengan menggunakan

framework Flask, integrasi dengan OpenAI API untuk analisis CVE, serta menggunakan Tailwind CSS untuk desain antarmuka pengguna.

3.2.1 Kebutuhan Fungsioanal

a. Autentikasi Pengguna: Sistem harus dapat melakukan registrasi, login, dan logout pengguna.
b. Pengelolaan Data CVE: Sistem harus dapat menyimpan, menampilkan, dan menganalisis data CVE.
c. Interaksi Pengguna: Sistem harus menyediakan fitur chat interaktif untuk analisis CVE dan pencarian informasi.
d. *Upload* CSV: Sistem harus mendukung upload file CSV untuk batch processing data CVE.
e. Dasbor: Sistem harus menyediakan Dasbor yang menampilkan statistik dan analisis data CVE.

3.2.2 Kebutuhan Non-Fungsional

a. Keamanan: Sistem harus memastikan keamanan data pengguna dan informasi CVE yang disimpan.
b. Proses Ekstraksi Detail CVE: Mengekstrak detail informasi dari data CVE.
c. Kinerja: Sistem harus mampu menangani beberapa pengguna secara bersamaan tanpa penurunan kinerja.
d. Antarmuka Pengguna: Sistem harus memiliki antarmuka pengguna yang intuitif dan mudah digunakan.
e. Pemeliharaan: Sistem harus mudah dipelihara dan dikembangkan lebih lanjut.

3.2.3 Analisis Kebutuhan Output

Keluaran yang dihasilkan sistem ini mencakup:

a. Otentikasi Pengguna:
    1. *Output***:** Pesan status pendaftaran dan login pengguna.
    2. Detail**:** Pesan konfirmasi untuk tindakan pendaftaran, *login*, dan *logout* yang berhasil.
b. Manajemen Thread:
    1. *Output***:** Pesan status pembuatan, penyimpanan, dan pengambilan *thread*.
    2. Detail**:** Pesan yang mengkonfirmasi pembuatan *thread* baru, penyimpanan riwayat sesi, dan pengambilan detail *thread*.

c. Penanganan Data CVE:
    1. *Output***:** Pesan status untuk pengambilan, penyimpanan, dan analisis data CVE.
    2. Detail**:** Konfirmasi operasi pengambilan dan penyimpanan data CVE yang berhasil, serta hasil analisis data CVE.
d. Pencarian dan Analisis:
    1. *Output***:** Hasil pencarian web dan analisis yang diringkas.
    2. Detail**:** Respon JSON yang berisi hasil pencarian web dan ringkasan informasi relevan.
e. Pencarian dan Analisis:
    1. *Output***:** Konten dinamis berdasarkan interaksi pengguna.
    2. Detail**:** Pembaruan real-time pada halaman dasbor, perpustakaan, dan thread berdasarkan tindakan pengguna.

3.2.4 Analisis Kebutuhan Tampilan Antarmuka

Antarmuka pengguna yang diperlukan adalah sebagai berikut:

a. Halaman *Login*.
    1. Form input untuk *email* dan *password*.
    2. Tombol untuk *submit form login*.
    3. Link untuk pendaftaran pengguna baru.
    4. Pesan error untuk login yang gagal.
b. Halaman Pendaftaran.
    1. Form input untuk nama pengguna, *email*, dan *password.*
    2. Tombol untuk submit form pendaftaran.
    3. Link untuk kembali ke halaman *login*.
    4. Pesan konfirmasi untuk pendaftaran yang berhasil.
c. Halaman Dasbor
    1. Tabel yang menampilkan daftar entri CVE dengan kolom untuk ID CVE, tingkat keparahan, deskripsi, tanggal publikasi, dan skor CVSS.
    2. Tombol untuk menambahkan entri CVE baru.
    3. Link navigasi untuk menuju halaman perpustakaan dan halaman thread.
d. Halaman *Library*
    1. Daftar *thread* yang dibuat oleh pengguna saat ini.
    2. Tombol untuk membuat thread baru.

3. *Link* untuk melihat detail masing-masing *thread*.
4. Tombol untuk menghapus *thread*.

e. Halaman *Thread*
   1. Judul *thread*.
   2. Riwayat sesi percakapan dalam *thread*.
   3. Form input untuk menambahkan pesan baru ke *thread*.
   4. Tombol untuk mengirim pesan baru.

f. Halaman Utama
   1. Form *input* untuk ID CVE.
   2. Tombol untuk mengirim permintaan detail CVE.
   3. Tampilan detail CVE yang mencakup ID CVE, tingkat keparahan, deskripsi, tanggal publikasi, tanggal modifikasi terakhir, skor CVSS, dan referensi.
   4. Form input untuk kueri pencarian.
   5. Tombol untuk mengirim permintaan pencarian.
   6. Tampilan hasil pencarian yang mencakup judul dan cuplikan dari hasil pencarian.

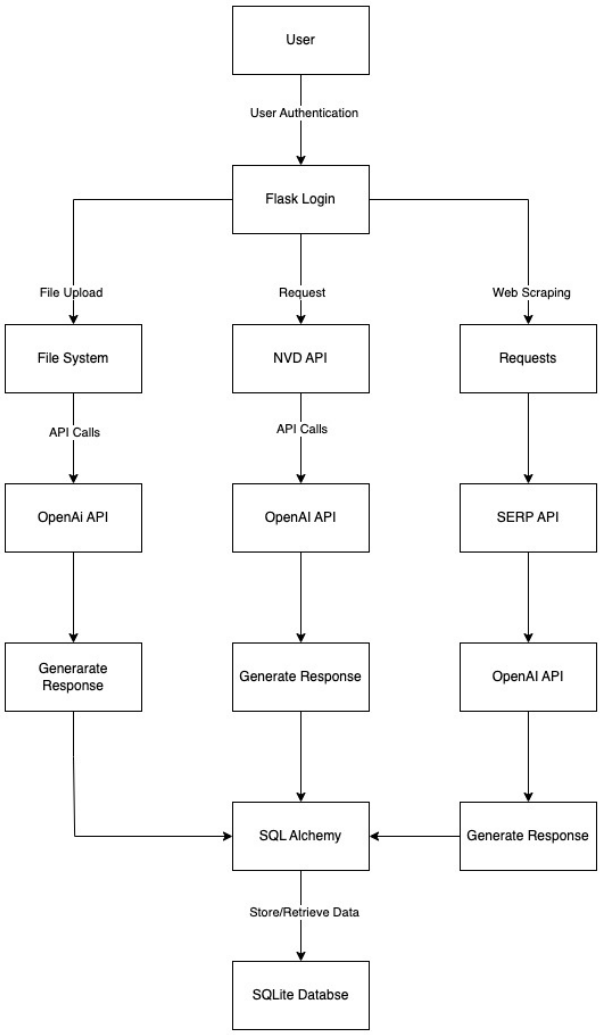
Antarmuka pengguna dirancang untuk menyediakan navigasi yang mudah dan akses cepat ke fungsi utama aplikasi, seperti otentikasi pengguna, manajemen *thread*, penanganan data CVE, dan pencarian informasi.

3.3 Perancangan Desain Sistem

Desain sistem ini menjelaskan bagaimana sistem dikembangkan untuk mengumpulkan data CVE, menganalisisnya menggunakan OpenAI API, dan menyajikan hasilnya kepada pengguna. Bagian ini mencakup diagram alir umum sistem, *use case diagram*, *activity diagram*, dan perencanaan perancangan basis data yang terdiri dari Entity Relationship Diagram (ERD) serta penjelasan tabel-tabel yang terlibat. Berikut adalah detail dari masing-masing bagian:

3.3.1 Gambaran Umum Sistem

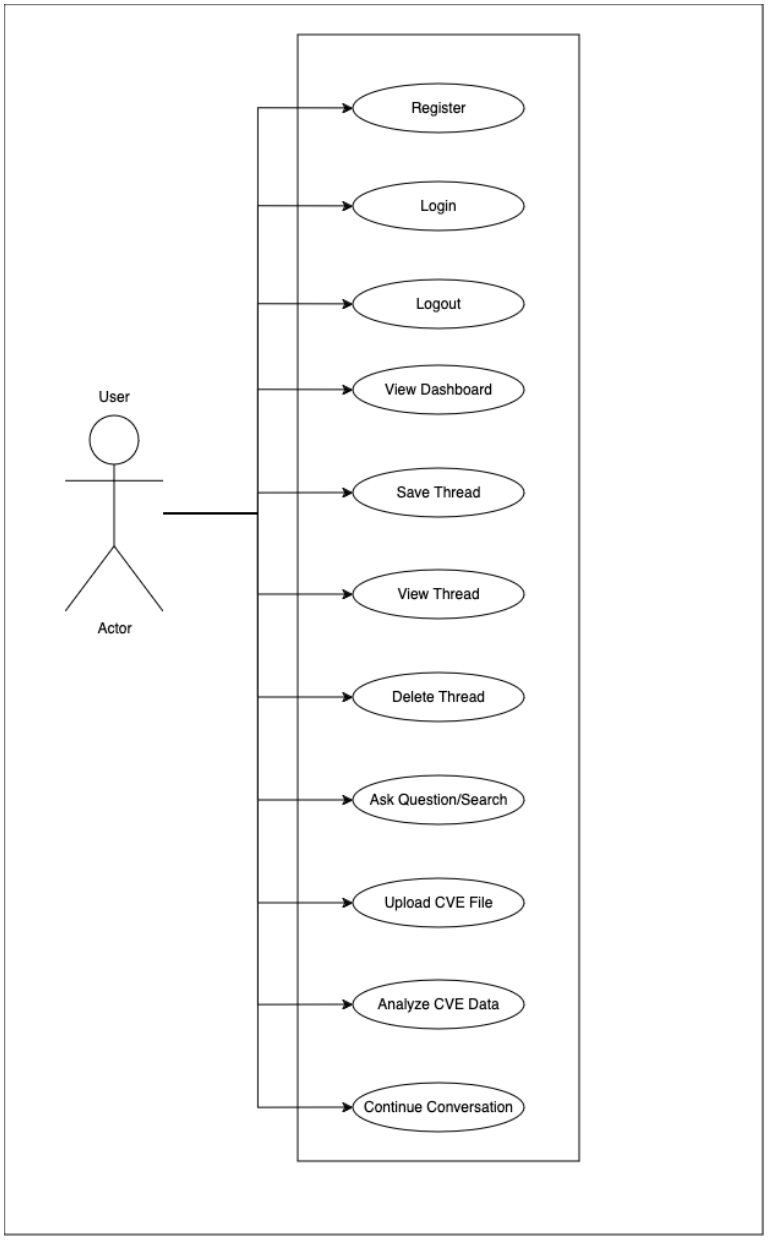
Sistem ini dirancang untuk mengumpulkan data CVE, menganalisisnya menggunakan OpenAI API, dan menampilkan hasil analisis kepada pengguna. Berikut adalah diagram alir umum dari sistem yang dikembangkan:



Gambar 3.1 Diagram Alir Umum Sistem

3.3.2 Use Case Diagram

Use case diagram menggambarkan interaksi antara aktor (pengguna) dengan sistem. Berikut adalah diagram use case untuk sistem yang dikembangkan.



Gambar 3.2 Use Case Diagram

Tabel 3.1Penjelasan Use case Diagram

No	**Aktor**	**Use Case**	**Deskripsi**
1	*User*	Registrasi	Pengguna baru mendaftar ke system dengan membuat akun baru.
2	*User*	*Login*	Pengguna masuk ke dalam system menggunakan kredensial yang telah terdaftar
3	*User*	*Logout*	Pengguna keluar dari system setelah selesai menggunakan aplikasi

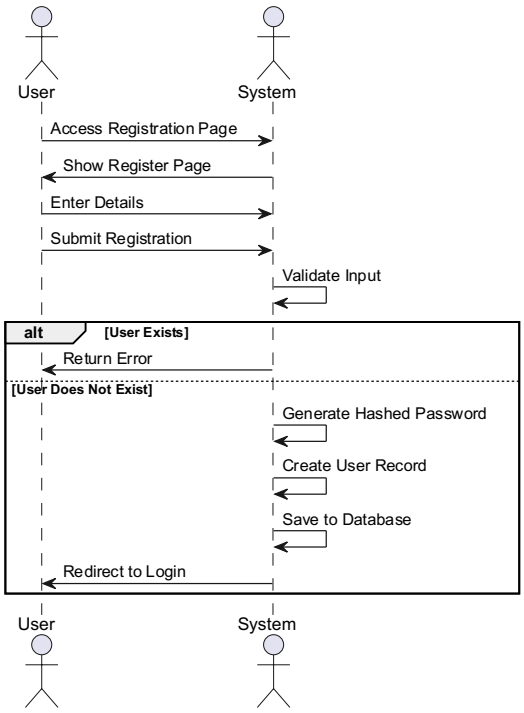
No	Aktor	Use Case	Deskripsi
4	*User*	View Dasbor	Pengguna melihat Dasbor yang berisi data CVE
5	*User*	*Save Thread*	Pengguna menyimpan sebuah thread (percakapan atau diskusi) ke dalam sistem untuk referensi atau akses di masa mendatang.
6	*User*	*View Thread*	Pengguna melihat atau mengakses sebuah thread yang telah disimpan atau ada di dalam sistem.
7	*User*	*Delete Thread*	Pengguna menghapus sebuah thread yang tidak lagi dibutuhkan atau diinginkan dari sistem.
8	*User*	*Ask Question/Search*	Pengguna mengajukan pertanyaan atau melakukan pencarian informasi dalam sistem, mungkin terkait CVE atau diskusi yang sedang berlangsung.
9	*User*	*Upload CVE File*	Pengguna mengunggah file CVE (Common Vulnerabilities and Exposures) dengan ekstensi CSV untuk dianalisis oleh sistem, mungkin untuk melihat kerentanan atau risiko yang terkait.
10	*User*	*Analyze CVE Data*	Pengguna meminta sistem untuk menganalisis data CVE yang telah diunggah, memberikan hasil analisis tentang kerentanan yang ditemukan.
11	*User*	*Continue Conversation*	Pengguna melanjutkan percakapan yang sedang berlangsung, mungkin untuk mengajukan pertanyaan lanjutan atau mendapatkan informasi lebih lanjut.

3.3.3 Sequence Diagram

Berikut adalah *sequence* diagram yang menggambarkan interaksi user dengan object-object yang terkait di dalam sistem:

a. Proses *Register*

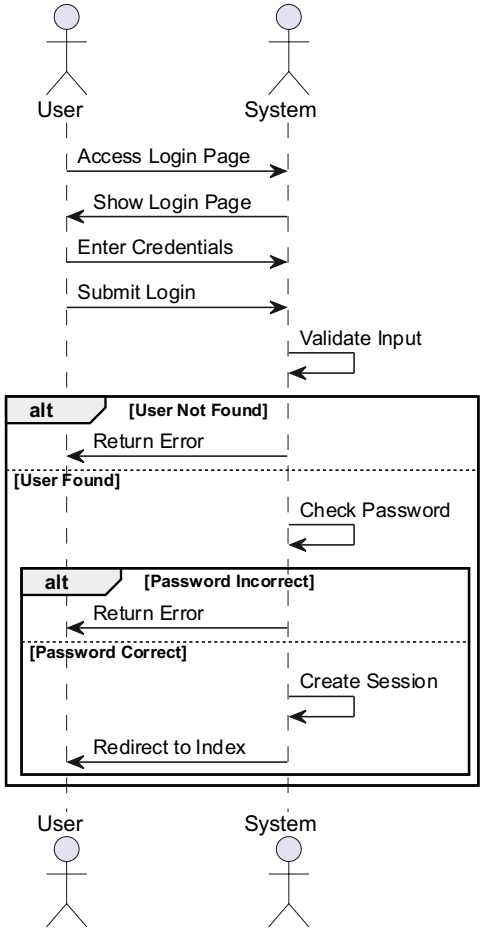
Gambar 3.3 menggambarkan proses registrasi pengguna baru dalam sistem. Proses ini melibatkan langkah-langkah seperti pengisian formulir pendaftaran, validasi data, dan penyimpanan informasi pengguna ke dalam database.



Gambar 3.3 Proses *Register*

b. Proses *Login*

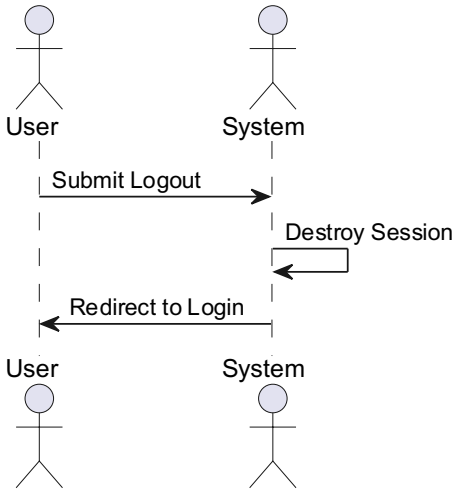
Gambar 3.4 menunjukkan proses login di mana pengguna memasukkan kredensial mereka (*username* dan *password*) untuk mendapatkan akses ke sistem. Proses ini termasuk validasi kredensial dan pemberian akses jika data yang dimasukkan benar.



Gambar 3.4 Proses *Login*

c. Proses *LogOut*

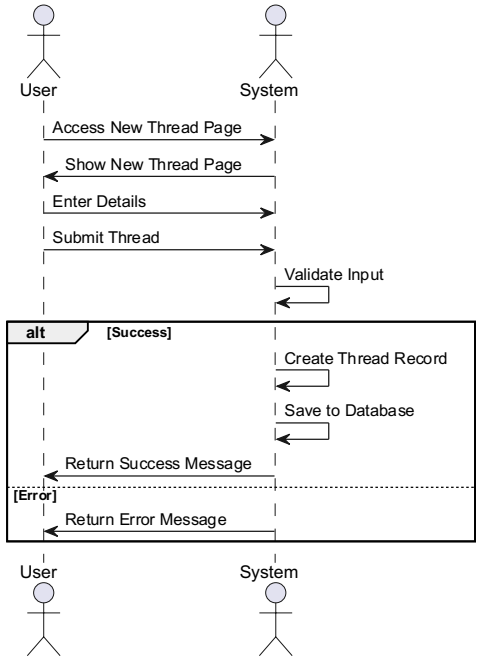
Gambar 3.5 menjelaskan proses logout yang dilakukan oleh pengguna untuk keluar dari sistem. Proses ini mencakup penghapusan sesi pengguna dan pengembalian pengguna ke halaman login atau halaman utama.



Gambar 3.5 Proses *LogOut*

d. Proses *Creating a New Thread*

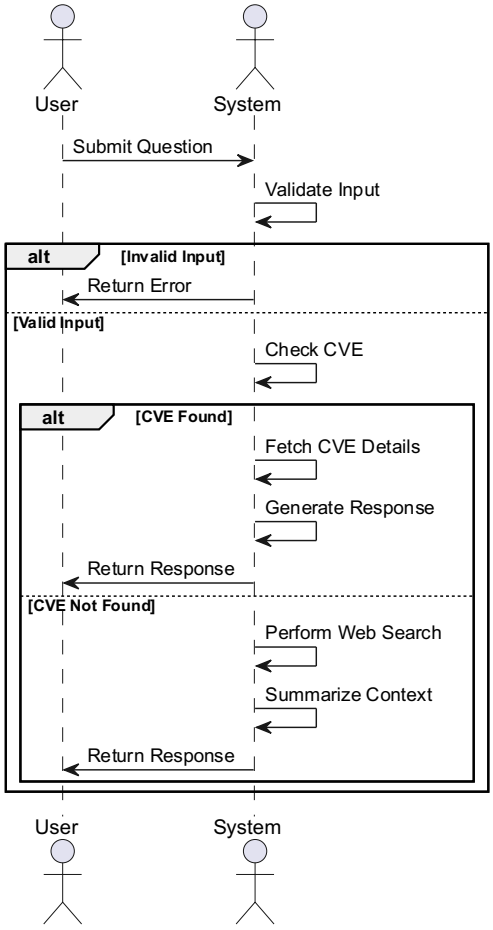
Gambar 3.6 menggambarkan proses pembuatan thread baru oleh pengguna dalam sistem. Langkah-langkah dalam proses ini meliputi pengisian detail thread, pengajuan thread, dan penyimpanan thread ke dalam database.



Gambar 3.6 Proses *Creating A New Thread*

e. Proses *Asking Question*

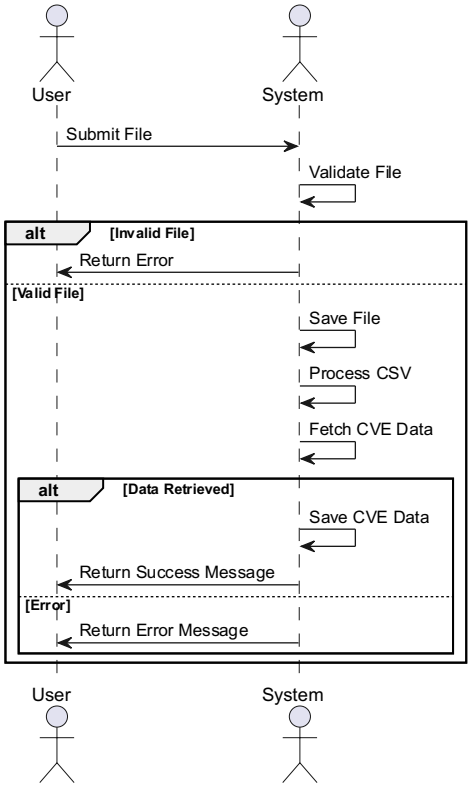
Gambar 3.7 menunjukkan proses pengajuan pertanyaan oleh pengguna. Proses ini melibatkan penulisan pertanyaan, pengiriman pertanyaan, dan penerimaan pertanyaan oleh sistem untuk direspon.



Gambar 3.7 Proses *Asking Question*

f. Proses *Uploading File*

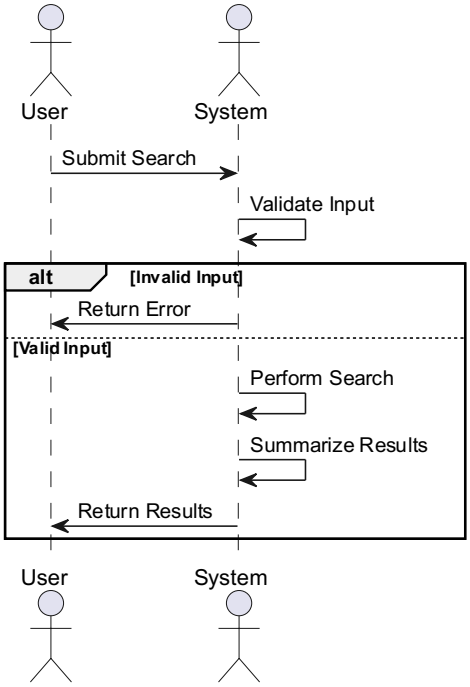
Gambar 3.8 menjelaskan proses unggah file di mana pengguna memilih file dari perangkat mereka dan mengunggahnya ke sistem. Langkah-langkahnya termasuk pemilihan file, validasi file, dan penyimpanan file ke server.



Gambar 3.8 Proses *uploading a file*

g. Proses *Web Search Process*

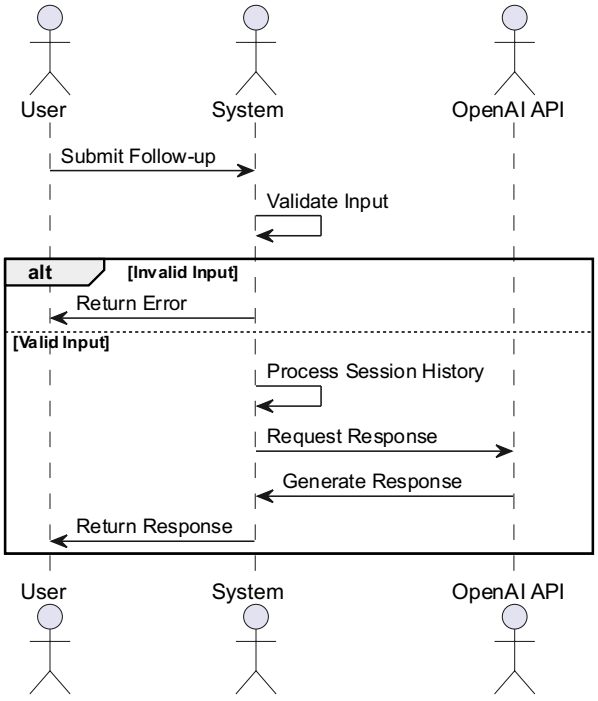
Gambar 3.9 menggambarkan proses pencarian web menggunakan SERP API di mana pengguna memasukkan kata kunci pencarian dan sistem melakukan pencarian berdasarkan kata kunci tersebut. Hasil pencarian kemudian ditampilkan kepada pengguna.



Gambar 3.9 Proses *web search process*

h. Proses *follow-up question*

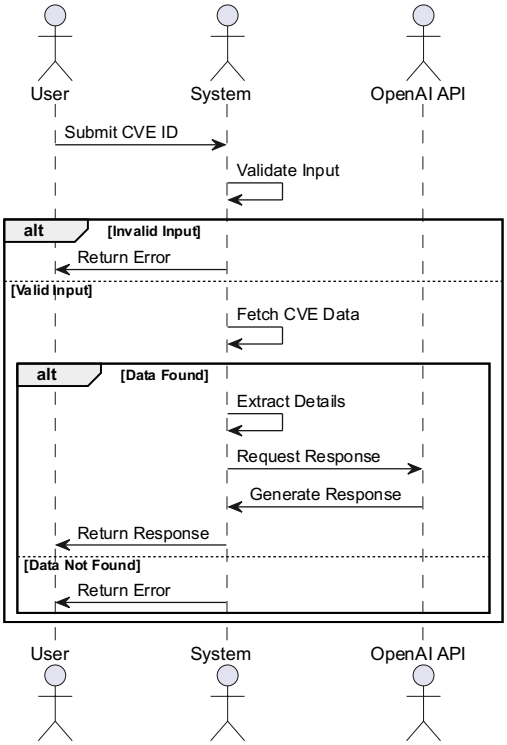
Gambar 3.10 menunjukkan proses pengajuan pertanyaan lanjutan oleh pengguna setelah menerima jawaban awal. Proses ini mencakup penulisan pertanyaan lanjutan dan pengiriman pertanyaan ke sistem untuk dijawab kembali.



Gambar 3.10 Proses *follow-up question*

i. Proses *extracting CVE data and generating an OpenAI response*

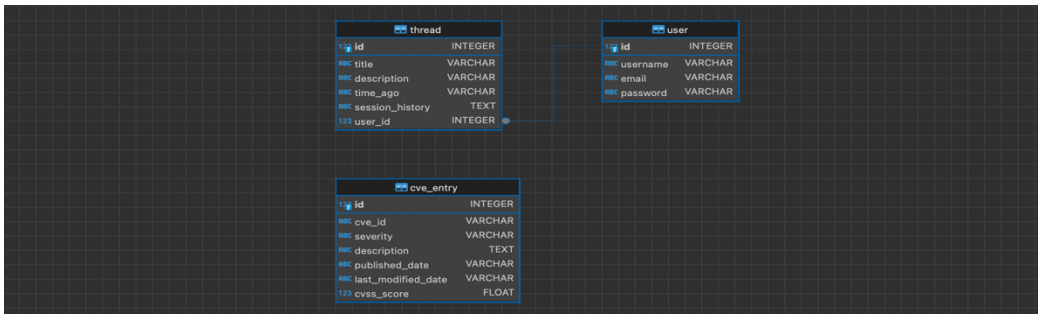
Gambar 3.11 menjelaskan proses ekstraksi data CVE dan pembuatan respons oleh OpenAI. Proses ini melibatkan pengambilan data CVE dari database, pengolahan data oleh OpenAI, dan penyajian hasil analisis kepada pengguna.



Gambar 3. 11 Proses *extracting CVE and generating an OpenAI response*

3.3.4 Perancangan Entity Relationship Diagram (ERD)

Entity Relationship Diagram menggambarkan hubungan antara entitas dalam basis data. ERD sistem ini meliputi tabel pengguna, thread, dan entri CVE.



Gambar 3. 12 Perancangan ERD

a. Tabel *User* :
    1. Id (Integer): Id pengguna (Primary Key)
    2. Username (String): Nama Pengguna
    3. Email (String): Email Pengguna
    4. Password (String): Password pengguna (hashed)
b. Tabel *Thread* :
    1. Id (Integer): Id thread (Primary Key)
    2. Title (String): Judul thread
    3. Time_ago (String): Waktu pembuatan thread relatif terhadap waktu sekarang
    4. Session_history (Text): Riwayat sesi yang terkait dengan thread
    5. Use_id (Integer): Id pengguna yang membuat thread (Foreign Key yang merujuk ke kolom `id` dalam table `User`
    6. last_modified_date (String): Tanggal modifikasi terakhir CVE
    7. cvss_score (Float) Skor CVSS CVE
c. Tabel CVEEntry:
    1. Id (Integer): Id thread (Primary Key)
    2. Cve_id (String): Id CVE yang Unik
    3. Severity (String): Tingkat keparahan CVE
    4. Description (Text): Deskripsi CVE
    5. Publised_date (String): Tanggal publikasi CVE
    6. Last_modified_date (String): Tanggal Modifikasi terakhir CVE
    7. Cvss_score (Float): Skor CVSS CVE

Tabel *User* memiliki relasi satu-ke-banyak (one-to-many) dengan tabel Thread, yang berarti seorang pengguna (User) dapat membuat banyak thread (Thread). Relasi ini diimplementasikan dengan kolom user_id dalam tabel Thread, yang merupakan foreign key yang merujuk ke kolom id dalam tabel User. Hal ini memungkinkan kita melacak siapa yang membuat setiap thread. Di sisi lain, tidak ada relasi langsung antara tabel Thread dan CVEEntry dalam model database yang diberikan. Tabel Thread digunakan untuk menyimpan informasi tentang thread yang dibuat oleh pengguna, sedangkan tabel CVEEntry menyimpan informasi tentang entri CVE. Namun, setiap thread mungkin mendiskusikan berbagai entri CVE berdasarkan konteksnya. Relasi antara User dan Thread memungkinkan kita memahami

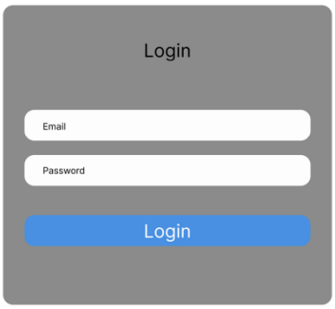
struktur data dan bagaimana data pengguna dan thread dihubungkan dalam sistem, meskipun tidak ada foreign key atau relasi langsung antara tabel Thread dan CVEEntry dalam model ini.

3.3.5 Perancangan Desain Antarmuka Pengguna

Perancangan antarmuka pengguna dilakukan dengan menggunakan TailwindCSS untuk memastikan tampilan yang responsif dan *user-friendly*. Berikut adalah desain awal untuk beberapa halaman penting dalam sistem.

a. Halaman *Login*

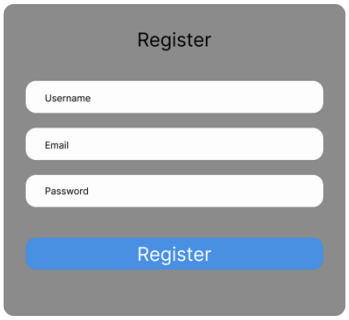
Gambar 3.13 menggambarkan halaman login di mana pengguna memasukkan kredensial mereka (*username* dan *password*) untuk mengakses sistem. Desain halaman ini dibuat sederhana dan intuitif untuk memudahkan pengguna dalam proses login.



Gambar 3.13 Halaman *Login*

b. Halaman Registrasi

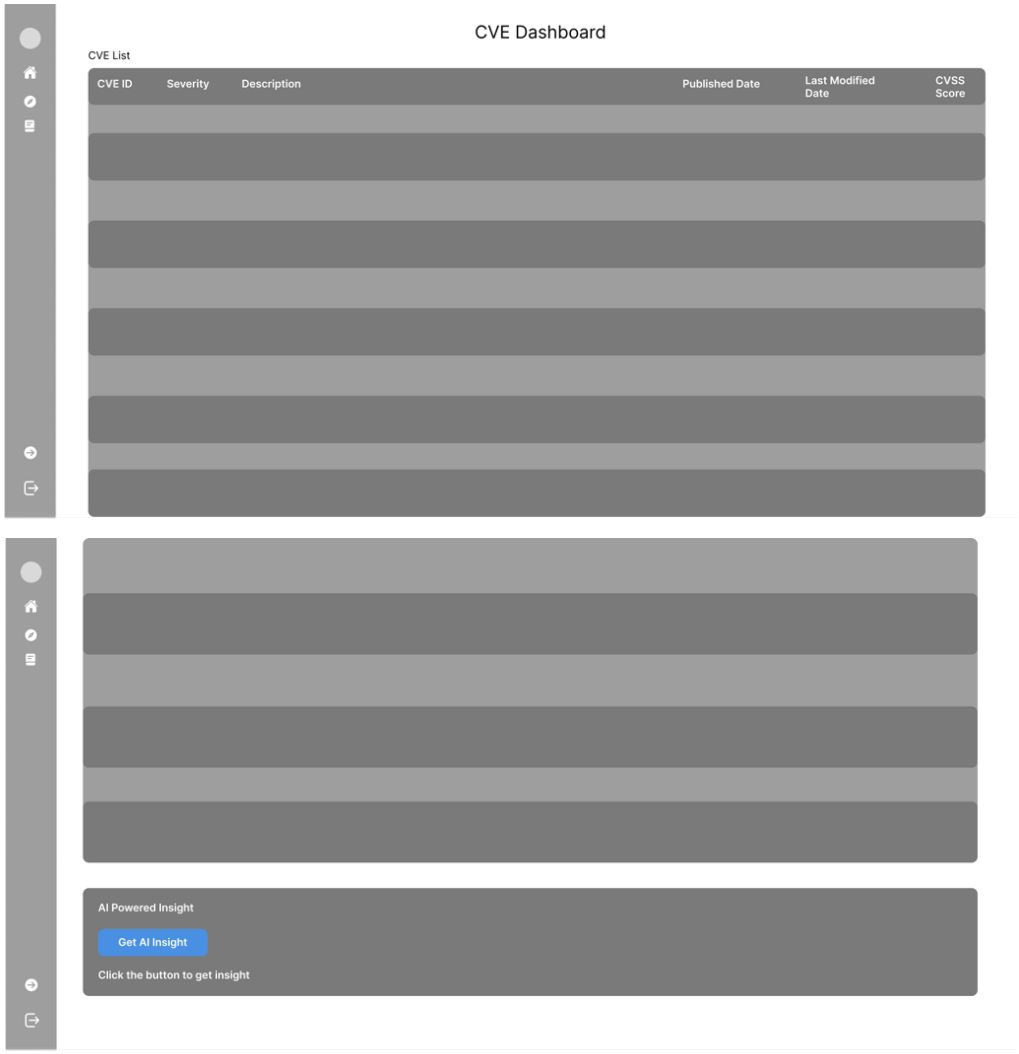
Gambar 3.14 menunjukkan halaman registrasi di mana pengguna baru dapat membuat akun dengan mengisi formulir pendaftaran. Formulir ini mencakup field seperti username, email, dan password. Desain halaman ini fokus pada kemudahan pengisian data dan panduan yang jelas untuk pengguna baru.



Gambar 3. 14 Halaman Registrasi

c. Halaman Dasbor

Gambar 3.15 menggambarkan halaman dasbor yang berisi daftar CVE yang telah di input user. Desain dasbor ini mengutamakan kenyamanan navigasi dan akses cepat ke informasi yang relevan.



Gambar 3. 15 Halaman Dasbor

d. Halaman Utama

Gambar 3.16 menunjukkan halaman utama yang berfungsi sebagai beranda sistem. Halaman ini memberikan pengenalan umum tentang sistem, fitur-fitur utama. Desain halaman utama dirancang untuk menarik perhatian pengguna dan memberikan informasi secara ringkas dan jelas.

Gambar 3. 16 Halaman utama

e. Halaman *Library*

Gambar 3.17 menggambarkan halaman library di mana pengguna dapat mengakses thread yang sudah disimpan sebelum nya untuk melihat riwayat percakapan. Desain halaman ini menekankan pada kemudahan pencarian dan pengorganisasian resource agar pengguna dapat dengan mudah menemukan dan mengakses riwayat yang mereka butuhkan.

Gambar 3. 17 Halaman *Library*

3.4 Implementasi Sistem

Implementasi sistem dilakukan dengan menggunakan berbagai teknologi yang terintegrasi untuk memenuhi kebutuhan fungsional dan non-fungsional. Berikut adalah penjelasan mengenai implementasi beberapa fitur utama:

3.4.1 Autentikasi Pengguna

Autentikasi pengguna diimplementasikan menggunakan Flask-Login untuk mengelola sesi pengguna dengan mudah. Dengan menggunakan Flask-Login, aplikasi dapat mengelola proses otentikasi, termasuk proses login dan logout pengguna. Ini memberikan keamanan tambahan dengan cara menyimpan informasi otentikasi dalam sesi yang dienkripsi, yang memungkinkan pengguna untuk tetap masuk secara aman dan nyaman selama sesi mereka menggunakan aplikasi.

3.4.2 Pengelolaan Data CVE

Data CVE dikelola menggunakan SQLAlchemy untuk interaksi dengan database. Data yang disimpan meliputi id CVE, tingkat keparahan, deskripsi, tanggal publikasi, tanggal modifikasi terakhir, dan skor CVSS.

3.4.3 Ekstraksi Data CVE

Data CVE diekstrak dari JSON yang diperoleh dari NVD. Berikut adalah struktur data JSON CVE dan penjelasan komponennya:

```
    {
        "resultsPerPage": 1,
        "startIndex": 0,
        "totalResults": 1,
        "format": "NVD_CVE",
        "version": "2.0",
        "timestamp": "2024-06-19T11:52:59.437",
        "vulnerabilities": [
            {
                "cve": {
                    "id": "CVE-2024-36837",
                    "sourceIdentifier":  "cve@mitre####  Struktur  Data
JSON CVE
```

Data CVE yang diperoleh dalam format JSON dari NVD terdiri dari beberapa komponen utama yang dapat dikelompokkan sebagai berikut:

1. **Metadata**: Bagian ini memberikan informasi umum tentang hasil pencarian.
   - **resultsPerPage**: Jumlah hasil yang ditampilkan per halaman.
   - **startIndex**: Indeks awal dari hasil yang ditampilkan.
   - **totalResults**: Total jumlah hasil yang ditemukan.
   - **format**: Format data yang digunakan (misalnya, "NVD_CVE").
   - **version**: Versi dari format data.
   - **timestamp**: Waktu pengambilan data dalam format ISO 8601.

2. **Vulnerabilities**: Bagian ini berisi informasi detail tentang kerentanan yang ditemukan.
   - **cve**: Objek yang berisi informasi tentang CVE tertentu:
     - **id**: ID unik dari CVE.
     - **sourceIdentifier**: Sumber yang mengidentifikasi CVE.
     - **published**: Tanggal publikasi CVE dalam format ISO 8601.
     - **lastModified**: Tanggal modifikasi terakhir CVE dalam format ISO 8601.
     - **vulnStatus**: Status analisis kerentanan.
     - **descriptions**: Daftar deskripsi CVE dalam berbagai bahasa.
       - **lang**: Bahasa deskripsi.
       - **value**: Deskripsi rinci tentang CVE.
     - **metrics**: Objek yang berisi metrik CVSS v3.1.
       - **cvssMetricV31**: Daftar metrik CVSS v3.1.
         - **source**: Sumber metrik.
         - **type**: Jenis metrik.
         - **cvssData**: Data CVSS:
           - **version**: Versi CVSS.
           - **vectorString**: String vektor CVSS.
           - **attackVector**: Vektor serangan.
           - **attackComplexity**: Kompleksitas serangan.
           - **privilegesRequired**: Privileges yang diperlukan.
           - **userInteraction**: Interaksi pengguna yang diperlukan.
           - **scope**: Lingkup.
           - **confidentialityImpact**: Dampak pada kerahasiaan.
           - **integrityImpact**: Dampak pada integritas.

```
            - **availabilityImpact**: Dampak pada ketersediaan.
            - **baseScore**: Skor dasar.
            - **baseSeverity**: Tingkat keparahan dasar.
            - **exploitabilityScore**: Skor eksploitasi.
            - **impactScore**: Skor dampak.
        - **weaknesses**: Daftar kelemahan yang terkait dengan CVE.
          - **source**: Sumber kelemahan.
          - **type**: Jenis kelemahan.
          - **description**: Deskripsi kelemahan.
            - **lang**: Bahasa deskripsi.
      - **
```

Gambar 3. 18 Ekstraksi data CVE

3.4.4 Struktur Data JSON CVE

Struktur data JSON CVE terdiri dari beberapa komponen utama:

a. *resultsPerPage*: Menunjukkan jumlah hasil yang ditampilkan per halaman (misalnya,
b. *startIndex*: Menunjukkan indeks awal dari hasil yang ditampilkan (misalnya, 0).
c. *totalResults*: Menunjukkan total jumlah hasil yang ditemukan (misalnya, 1).
d. format: Format data yang digunakan (misalnya, "NVD_CVE").
e. *version*: Versi dari format data (misalnya, "2.0").
f. *timestamp*: Waktu pengambilan data dalam format ISO 8601 (misalnya, "2024-06-19T11:52:59.437").

3.4.5 Bagian Vulnerabilities

Bagian ini berisi informasi detail tentang kerentanan yang ditemukan:

a. resultsPerPage: Menunjukkan jumlah hasil yang ditampilkan per halaman

b. CVE: Objek yang berisi informasi tentang CVE tertentu:

   1. id: ID unik dari CVE (misalnya, "CVE-2024-36837").
   2. sourceIdentifier: Sumber yang mengidentifikasi CVE (misalnya, "cve@mitre.org").
   3. published: Tanggal publikasi CVE dalam format ISO 8601 (misalnya, "2024-06-05T15:15:11.803").
   4. lastModified: Tanggal modifikasi terakhir CVE dalam format ISO 8601 (misalnya, "2024-06-18T18:54:51.380").

5. vulnStatus: Status analisis kerentanan (misalnya, "Analyzed").
6. descriptions: Daftar deskripsi CVE dalam berbagai bahasa:
    - lang: Bahasa deskripsi (misalnya, "en" untuk Inggris).
    - value: Deskripsi rinci tentang CVE (misalnya, "SQL Injection vulnerability in CRMEB v.5.2.2 ...").
7. Metrics: Objek yang berisi metrik CVSS v3.1:
8. CvssMetricV31: Daftar metrik CVSS v3.1:
    - source: Sumber metrik (misalnya, "nvd@nist.gov").
    - type: Jenis metrik (misalnya, "Primary").
    - cvssData: Data CVSS:
    - version: Versi CVSS (misalnya, "3.1").
    - vectorString: String vektor CVSS (misalnya, "CVSS:3.1/AV/AC/PR/UI/S/C/I/A").
    - attackVector: Vektor serangan (misalnya, "NETWORK").
    - attackComplexity: Kompleksitas serangan (misalnya, "LOW").
    - privilegesRequired: Privileges yang diperlukan (misalnya, "NONE").
    - userInteraction: Interaksi pengguna yang diperlukan (misalnya, "NONE").
    - scope: Lingkup (misalnya, "UNCHANGED").
    - confidentialityImpact: Dampak pada kerahasiaan (misalnya, "HIGH").
    - integrityImpact: Dampak pada integritas (misalnya, "NONE").
    - availabilityImpact: Dampak pada ketersediaan (misalnya, "NONE").
    - baseScore: Skor dasar (misalnya, 7.5).
    - baseSeverity: Tingkat keparahan dasar (misalnya, "HIGH").
    - exploitabilityScore: Skor eksploitasi (misalnya, 3.9).
    - impactScore: Skor dampak (misalnya, 3.6).
    - weaknesses: Daftar kelemahan yang terkait dengan CVE:
    - source: Sumber kelemahan (misalnya, "nvd@nist.gov").

- type: Jenis kelemahan (misalnya, "Primary").
- description: Deskripsi kelemahan:
- lang: Bahasa deskripsi (misalnya, "en").
- value: Nilai deskripsi kelemahan (misalnya, "CWE-89").
- configurations: Konfigurasi yang terpengaruh oleh CVE:
- nodes: Daftar node konfigurasi:
- operator: Operator logis (misalnya, "OR").
- negate: Apakah kondisi dinafikkan (misalnya, false).
- cpeMatch: Daftar kecocokan CPE:
- vulnerable: Apakah rentan (misalnya, true).
- criteria: Kriteria CPE (misalnya, "cpe:2.3:a:crmeb:crmeb:5.2.2:::::::*").
- matchCriteriaId: ID kriteria kecocokan (misalnya, "F0566B98-D93C-4D4D-BB2B-37FCF55D0585").
- references: Daftar referensi yang terkait dengan CVE:
- url: URL referensi (misalnya, "https://github.com/phtcloud-dev/CVE-2024-36837").
- source: Sumber referensi (misalnya, "cve@mitre.org").
- tags: Daftar tag referensi (misalnya, ["Third Party Advisory"]).

3.4.6 *Chat* Interaktif

Chat interaktif diimplementasikan menggunakan OpenAI API untuk memberikan analisis CVE berdasarkan input pengguna. Riwayat sesi chat dapat disimpan dalam database untuk referensi dan analisis lebih lanjut.

3.4.7 Detail Proses *Prompting*

Persiapan *Prompt***,** Sistem menyusun *prompt* yang berisi rincian lengkap dari CVE seperti ID, tingkat keparahan, deskripsi, tanggal publikasi, tanggal modifikasi terakhir, skor CVSS, string vektor, dan referensi. Contoh prompt:

```
Provide an in-depth analysis of the identified CVE:
1. CVE Identification:
   - CVE ID: CVE-2023-12345
   - Severity: High
   - Description: This is a sample CVE description.
   - Published Date: 2023-06-01
   - Last Modified Date: 2023-06-10
   - CVSS Score: 9.0
   - Vector String: AV:N/AC:L/Au:N/C:C/I:C/A:C
   - References: [{'url': 'http://example.com', 'tags': ['tag1',
'tag2']}]
```

Gambar 3. 19 Detail *Prompting* Rincian CVE

a. Penyesuaian Berdasarkan Audiens: Bagian tambahan ditambahkan ke *prompt* berdasarkan audiens yang dituju (pemula, mahir, atau umum). Contoh untuk pemula:

```
2. Impact:
   - Describe in detail terms how this CVE can affect vulnerable
systems. Include potential damage to systems, data, and operations.
3. Mitigation Steps:
   - Provide detailed steps to mitigate the risks. Include
recommendations on software updates, system configurations, and best
practices.
4. Recommendations for Organizations:
   - Provide in-depth advice for handling this vulnerability. Include
security policies, employee training, and monitoring strategies.
```

Gambar 3. 20 Contoh untuk pemula

b. Prompt ke Model**:** *Large Language Model* menghasilkan respons berdasarkan *prompt* dan riwayat sesi. Respons ini direview dan jika perlu, ditambahkan pertanyaan atau klarifikasi lanjutan untuk memperbaiki analisis:

```
prompt = ( f"\"\"\"\n" f"{introduction}" f"{cve_identification}"
f"{instructions}" f"{role_prompt}" f"{example}" f'\"\"\"' )
```

Gambar 3. 21 Prompt ke model

c. Penerimaan dan Pemrosesan Respons**:** Respons dari LLM diproses dan disimpan dalam riwayat sesi. Penyesuaian dilakukan berdasarkan kejelasan dan kelengkapan respons untuk memastikan output akhir komprehensif dan sesuai kebutuhan pengguna.

d. Penyimpanan dan Tampilan

Respons akhir dari model disimpan dalam riwayat sesi chat dan ditampilkan kepada pengguna dalam antarmuka yang mudah dimengerti. Hal ini memungkinkan pengguna untuk melihat seluruh percakapan yang telah terjadi, baik pesan dari pengguna maupun respons dari model AI, dalam format yang jelas dan terstuktur. Dengan menggunakan Flask sebagai *framework backend*, kita dapat menangani pengiriman pesan dan penyimpanan riwayat chat secara efisien. Di sisi frontend, TailwindCSS digunakan untuk mendesain antarmuka yang bersih dan responsif, sehingga memudahkan pengguna dalam berinteraksi dengan aplikasi chat.

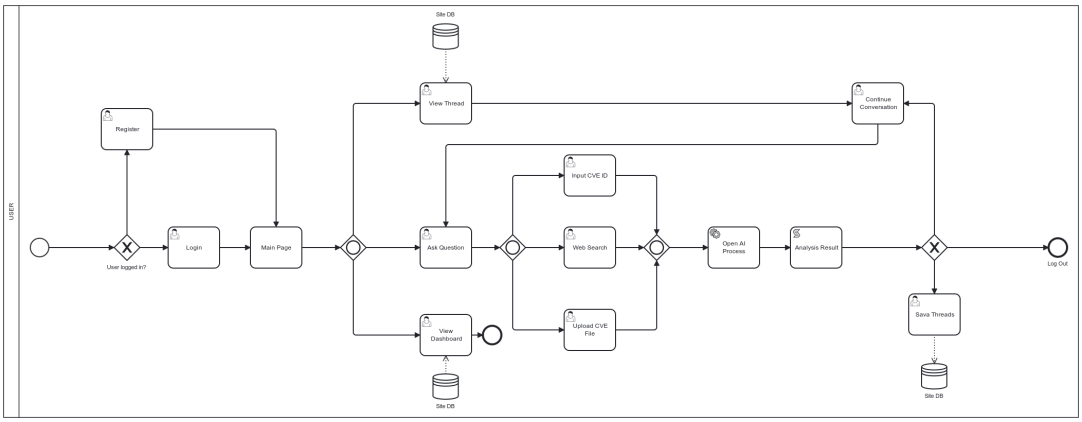
Pada bagian *backend*, setiap pesan yang dikirim oleh pengguna diterima melalui *endpoint* yang disediakan dan kemudian disimpan dalam sesi. Respons dari model AI kemudian diambil menggunakan API OpenAI dan juga disimpan dalam sesi yang sama. Seluruh riwayat percakapan ini kemudian dapat diambil kembali dan ditampilkan kepada pengguna saat mereka mengakses aplikasi.

Di sisi frontend, kita menyediakan antarmuka yang menampilkan riwayat percakapan dengan menggunakan TailwindCSS untuk memastikan tampilan yang menarik dan mudah digunakan. Setiap pesan dari pengguna dan respons dari model ditampilkan dalam bentuk balon chat, dengan styling yang membedakan antara pesan pengguna dan pesan sistem. Dengan demikian, pengguna dapat dengan mudah mengikuti alur percakapan dan mengerti respons yang diberikan oleh model AI.

BAB IV
HASIL DAN PEMBAHASAN

Bab ini melanjutkan proses yang telah dibahas pada bab sebelumnya, yaitu analisis kebutuhan dan desain sistem. Pada bab ini, akan diuraikan implementasi sistem yang telah dikembangkan beserta diagram proses bisnis nya, serta hasil dari pengujian dan pembahasan mengenai efektivitas sistem dalam mencapai tujuan yang diharapkan. Implementasi sistem ini mencakup struktur kode, konfigurasi, serta penjelasan mengenai komponen-komponen utama yang telah diterapkan.

4.1 Diagram Proses Bisnis



Gambar 4. 1 Diagram Proses Bisnis

Diagram proses bisnis yang ditampilkan pada Gambar 4.1 menggambarkan alur kerja sistem analisis CVE (Common Vulnerabilities and Exposures) berbasis web. Sistem ini dirancang untuk memberikan pengalaman yang komprehensif dan terintegrasi bagi pengguna dalam melakukan penelitian dan analisis kerentanan keamanan siber.

Proses dimulai dengan autentikasi pengguna, di mana sistem memeriksa status *login*. Pengguna yang belum memiliki akun dapat melakukan registrasi, sementara pengguna yang sudah terdaftar dapat langsung masuk ke sistem. Setelah berhasil login, pengguna diarahkan ke halaman utama yang menjadi pusat navigasi untuk berbagai fitur sistem.

Dari halaman utama, pengguna memiliki beberapa opsi utama. Mereka dapat melihat thread diskusi yang ada, yang diambil dari database sistem. Fitur ini memungkinkan pengguna untuk melanjutkan percakapan atau diskusi yang sedang berlangsung. Opsi lainnya adalah mengajukan pertanyaan baru, yang dapat dilakukan melalui input CVE ID spesifik atau

melakukan pencarian web. Sistem juga menyediakan dashboard yang menampilkan statistik atau ringkasan terkait CVE, dengan kemampuan untuk mengunggah file CVE.

Setelah pengguna memilih untuk mencari informasi atau menganalisis CVE tertentu, sistem melanjutkan ke tahap pemrosesan AI. Proses ini melibatkan analisis mendalam terhadap CVE atau hasil pencarian menggunakan teknologi kecerdasan buatan. Hasil analisis kemudian disajikan kepada pengguna, yang dapat memilih untuk melanjutkan percakapan, mengajukan pertanyaan lanjutan, atau menyimpan hasil analisis ke dalam database.

Seluruh proses ini didukung oleh integrasi database yang menyimpan informasi pengguna, data CVE, dan riwayat percakapan. Sistem ini dirancang untuk memberikan alur kerja yang lancar dan efisien, memungkinkan pengguna untuk dengan mudah mengakses, menganalisis, dan menyimpan informasi penting terkait kerentanan keamanan.

Diagram ini menunjukkan pendekatan holistik dalam manajemen dan analisis CVE, menggabungkan elemen-elemen kunci seperti autentikasi pengguna, pencarian informasi, analisis berbasis AI, dan penyimpanan data. Desain ini memungkinkan para profesional keamanan siber untuk bekerja secara efektif dalam mengidentifikasi, menganalisis, dan memahami kerentanan keamanan, sambil memfasilitasi kolaborasi dan berbagi pengetahuan melalui fitur thread dan percakapan berkelanjutan.

4.2 Penerapan

Sub-bab ini menjelaskan langkah-langkah penerapan sistem yang telah dikembangkan. Penerapan ini mencakup beberapa aspek penting seperti:

4.2.1 Struktur Kode

Struktur kode aplikasi Flask yang dikembangkan terdiri dari beberapa bagian penting:

a. File Utama (app.py): File ini berisi konfigurasi utama dan endpoint yang digunakan untuk menjalankan aplikasi. Di dalamnya terdapat pengaturan seperti konfigurasi database, pengaturan login, serta definisi endpoint yang akan diakses oleh pengguna.
b. Templates: Direktori ini berisi template HTML yang digunakan untuk tampilan antar pengguna, memungkinkan integrasi langsung dengan variable dan logika python.
c. Static Files: Direktori ini berisi file statis CSS dan JavaScript untuk. File-file ini digunakan untuk memberikan styling dan interaktivitas pada sisi klien.

4.2.2 Konfigurasi Flask dan OpenAI

Bagian ini menjelaskan bagaimana Flask dan OpenAI dikonfigurasi dalam aplikasi. Berikut adalah beberapa poin utama:

1. Flask Configuration: Aplikasi Flask diinisiasi dengan beberapa konfigurasi dasar seperti 'SECRET_KEY' untuk menjaga sesi pengguna dan 'SQLALCHEMY_DATABASE_URI' untuk menghubungkan aplikasi dengan database SQLite.
2. SQLAlchemy: Digunakan untuk interaksi dengan database. Model 'User', 'Thread', dan 'CVEEntry' didefinisikan untuk menyimpan data pengguna, thread percakapan, dan entri CVE.
3. Flask-Login: Digunakan untuk mengelola sesi pengguna, termasuk login dan logout. Fungsi 'load_user' digunakan untuk memuat objek pengguna berdasarkan ID.
4. OpenAI Configuration; API OpenAI dikonfigurasi dengan kunci API yang diambil dari variable lingkungan. API ini digunakan untuk menghasilkan respon dari model AI berdasarkan input pengguna.
5. Logging: Logging dikonfigurasikan untuk mencatat aktivitas aplikasi, membantu dalam debugging dan pemantauan.

Kode berikut menunjukkan bagaimana Flask dan OpenAI dikonfigurasi:

```
from flask import Flask, request, jsonify, render_template, redirect, url_for, session
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager, UserMixin, login_user, logout_user, current_user, login_required
from werkzeug.security import generate_password_hash, check_password_hash
import os
import requests
import pandas as pd
from openai import OpenAI
from datetime import datetime
import re
import logging
from flask import g

# Configure logging
```

```
logging.basicConfig(level=logging.INFO)

client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
app = Flask(__name__)
app.config['SECRET_KEY'] = 'your_secret_key'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///site.db'
db = SQLAlchemy(app)
login_manager = LoginManager(app)
login_manager.login_view = 'login'

session_history = []
threads = []  # Store threads
saved_threads = []

class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(150), unique=True, nullable=False)
    email = db.Column(db.String(150), unique=True, nullable=False)
    password = db.Column(db.String(150), nullable=False)
    threads = db.relationship('Thread', backref='author', lazy=True)

class Thread(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(200), nullable=False)
    description = db.Column(db.String(500), nullable=False)
    time_ago = db.Column(db.String(100), nullable=False)
    session_history = db.Column(db.Text, nullable=False)
    user_id      =      db.Column(db.Integer,      db.ForeignKey('user.id'),
nullable=False)

class CVEEntry(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    cve_id = db.Column(db.String(100), unique=True, nullable=False)
    severity = db.Column(db.String(50), nullable=False)
    description = db.Column(db.Text, nullable=False)
    published_date = db.Column(db.String(50), nullable=False)
    last_modified_date = db.Column(db.String(50), nullable=False)
    cvss_score = db.Column(db.Float, nullable=False)

@login_manager.user_loader
```

```
def load_user(user_id):
    return User.query.get(int(user_id))

@app.before_request
def before_request():
    g.session_history = []
```

Gambar 4. 2 Konfigurasi Flask dan OpenAI

4.2.3 Endpoint Utama

a. Endpoint Utama (/): Endpoint ini menampilkan halaman utama. Jika pengguna sudah login, mereka akan diarahkan ke halaman indeks, sedangkan jika belum login, mereka akan diarahkan ke halaman login.

```
@app.route('/')
def home():
    if 'user_id' in session:
        return redirect(url_for('index'))
    return redirect(url_for('login'))
```

Gambar 4. 3 Halaman Utama

b. Endpoint untuk Memulai Thread Baru (/new_thread): Endpoint ini digunakan untuk menginisiasi thread baru. Ketika endpoint ini diakses melalui metode POST, variable 'session_history' direset menjadi kosong untuk memulai sesi baru.

```
@app.route('/new_thread', methods=['POST'])
@login_required
def new_thread():
    global session_history
    session_history = []
    return jsonify({"message": "New thread started."})
```

Gambar 4. 4 Menginisiasi Thread Baru

c. Endpoint untuk Menampilkan Library (/library): Endpoint ini Menampilkan riwayat percakapan pengguna. Data Thread pengguna diambil dari database dan ditampilkan menggunakan 'library.html'.

```
@app.route('/library')
@login_required
def library():
    user_threads = Thread.query.filter_by(user_id=current_user.id).all()
    return render_template('library.html', threads=user_threads)
```

Gambar 4. 5 Menampilkan Riwayat Percakapan

d. Endpoint untuk Mengajukan Pertanyaan (/ask): Endpoint ini Mengelola input pengguna dan menghasilkan respon dari OpenAI. Jika input berisi ID CVE, aplikasi akan mengambil data CVE tersebut dan menghasilkan analisis. Jika tidak, aplikasi akan melakukan pencarian web dan meringkas hasilnya.

```
@app.route('/ask', methods=['POST'])
@login_required
def ask():
    data = request.get_json()
    if not data or 'user_input' not in data:
        return jsonify({"error": "Invalid input"}), 400

    user_input = data['user_input'].strip().upper()
    audience = data.get('audience', 'anyone')

    try:
        if "CVE-" in user_input:
            cve_id = user_input.split('/')[-1]
            result = fetch_cve_data(cve_id)
            if result and 'error' not in result:
                details = extract_details(result)
                if details:
                    openai_response= get_openai_response(details, audience)
                    logging.info("Saving CVE details to DB")
                    save_cve_to_db(details)
                    return jsonify({"response": openai_response})
                else:
                    return jsonify({"error": "Failed to extract CVE
details."})
            else:
                return jsonify({"error": result['error'] if result else "No
CVE found."})
        else:
```

```
            search_results = perform_web_search(user_input)
            if search_results:
                summarized_context = summarize_context(search_results)
                return   jsonify({"results":   search_results,   "summary":
summarized_context})
            else:
                return  jsonify({"results":  [],  "summary":  "No  relevant
information found."})
    except Exception as e:
        logging.error(f"Error in /ask endpoint: {str(e)}")
        return jsonify({"error": str(e)}), 500
```

Gambar 4. 6 Mengelola Input Pengguna Dan Menghasilkan Respon

e. Endpoint untuk Mengunggah File CSV (/upload): Endpoint ini Mengelola unggahan file CSV. File CSV yang diunggah diproses untuk mengambil ID CVE, dan data CVE tersebut kemudian diambil dan dianalis menggunakan OpenAI API.

```
@app.route('/upload', methods=['POST'])
@login_required
def upload():
    if 'file' not in request.files:
        return jsonify({"error": "No file part"}), 400

    file = request.files['file']
    if file.filename == '':
        return jsonify({"error": "No selected file"}), 400

    if file:
        file_path = os.path.join("/tmp", file.filename)
        file.save(file_path)
        cve_ids = process_csv(file_path)
        responses = []
        for cve_id in cve_ids:
            cve_id = cve_id.upper()
            result = fetch_cve_data(cve_id)
            if 'error' not in result:
                details = extract_details(result)
                if details:
                    openai_response = get_openai_response(details)
                    logging.info("Saving CVE details to DB")
```

```
                    save_cve_to_db(details)
                    responses.append({cve_id: openai_response})
                else:
                    responses.append({cve_id:   "Failed   to   extract   CVE
details."})
            else:
                responses.append({cve_id: result['error']})
        return jsonify({"responses": responses})
    return jsonify({"error": "File processing failed"}), 400
```

Gambar 4. 7 Mengelola Unggahan File CSV

f. Endpoint untuk Menindaklanjuti Pertanyaan (/follow-up): Endpoint ini Mengelola pertanyaan lanjutan dari pengguna. Pertanyaan lanjutan ditambahkan ke 'session_history' dan OpenAI menghasilkan respon berdasarkan riwayat sesi.

```
@app.route('/follow-up', methods=['POST'])
@login_required
def follow_up():
    try:
        data = request.get_json()
        if not data or 'follow_up_question' not in data:
            return jsonify({"error": "Invalid input"}), 400

        follow_up_question = data['follow_up_question'].strip()
        logging.info(f"Processing            follow-up            question:
{follow_up_question}")
        session_history.append({"role":          "user",          "content":
follow_up_question})

        response = client.chat.completions.create(
            model="gpt-4o",
            messages=session_history,
            max_tokens=100
        )
        follow_up_response = response.choices[0].message.content.strip()
        session_history.append({"role":       "assistant",       "content":
follow_up_response})

        logging.info(f"Follow-up response: {follow_up_response}")
        return jsonify({"response": follow_up_response})
    except Exception as e:
```

```
        logging.error(f"Error in /follow-up endpoint: {str(e)}")
        return jsonify({"error": str(e)}), 500
```

Gambar 4. 8 Mengelola Pertanyaan Lanjutan

4.2.4 Pengambilan Data CVE

a. Fungsi untuk mengambil Data CVE dari NVD: Fungsi ini mengambil data CVE dari layanan NVD (National Vulnerability Database) menggunakan permintaan HTTP. Jika permintaan berhasil, data CVE dikembalikan dalam format JSON.

```
def fetch_cve_data(cve_id):
    url =f"https://services.nvd.nist.gov/rest/json/cves/2.0?cveId={cve_id}"
    try:
        response = requests.get(url)
        response.raise_for_status()
        return response.json()
    except requests.RequestException as e:
        return {"error": f"Error fetching CVE data: {str(e)}"}
```

Gambar 4. 9 Fungsi untuk mengambil Data CVE

b. Fungsi untuk Mencari CVE menggunakan Google: Fungsi ini menggunakan API pencarian SERP untuk mencari informasi terkait CVE atau informasi lain di Google. Hasil pencarian dikembalikan dalam format JSON.

```
def perform_web_search(query):
    api_key = ''
    search_url = f"https://serpapi.com/search?q={query}&api_key={api_key}"
    try:
        response = requests.get(search_url)
        response.raise_for_status()
        data = response.json()
        return data['organic_results']
    except requests.RequestException as e:
        return {"error": f"Error accessing SERP API: {str(e)}"}
```

Gambar 4. 10 Mencari CVE menggunakan Google

4.2.5 Ekstraksi Detail CVE

a. Fungsi untuk mengekstrak detail CVE: Fungsi ini mengekstrak detail dari data CVE yang diambil dari NVD. Detail ini yang diekstrak meliputi ID CVE, tingkat keparahan, deskripsi, tanggal publikasi, tanggal modifikasi terakhir, skor CVSS, string vector, referensi, dan konfigurasi CPE.

```
def extract_details(cve_data):
    if 'vulnerabilities' in cve_data and cve_data['vulnerabilities']:
        vuln_data = cve_data['vulnerabilities'][0]
        if 'cve' in vuln_data:
            vuln = vuln_data['cve']
            details = {
                'id': vuln['id'],
                'severity':
vuln['metrics']['cvssMetricV31'][0]['cvssData']['baseSeverity']          if
'cvssMetricV31' in vuln['metrics'] else 'N/A',
                'description':     vuln['descriptions'][0]['value']     if
'descriptions' in vuln and vuln['descriptions'] else 'N/A',
                'published_date': vuln['published'] if 'published' in vuln
else 'N/A',
                'last_modified_date':        vuln['lastModified']        if
'lastModified' in vuln else 'N/A',
                'cvss_score':
vuln['metrics']['cvssMetricV31'][0]['cvssData']['baseScore']             if
'cvssMetricV31' in vuln['metrics'] else 'N/A',
                'vector_string':
vuln['metrics']['cvssMetricV31'][0]['cvssData']['vectorString']          if
'cvssMetricV31' in vuln['metrics'] else 'N/A',
                'references':     [{'url':     ref.get('url'),     'tags':
ref.get('tags', [])} for ref in vuln.get('references', [])],
                'cpe_configurations':
extract_cpe_details(vuln['configurations']) if 'configurations' in vuln else
[]
            }
            return details
    return None
```

Gambar 4. 11 Fungsi Untuk Mengekstrak Detail CVE

4.2.6 Integrasi OpenAI untuk Analisis Data

a. Fungsi untuk mengambil respon dari Open AI: Fungsi ini menghasilkan prompt berdasarkan detail CVE dan mengirimkannya ke OpenAI untuk mendapatkan analisis mendalam. Prompt disesuaikan berdasarkan audiens (pemula, lanjutan, atau umum) dan mencakup identifikasi CVE, dampak, langkah mitigasi, dan rekomendasi untuk organisasi.

```
    def def get_openai_response(details, audience='anyone'):
        # Introduction
        introduction = (
            f"\"\"\"\n"
            f"### CVE Analysis Request ###\n"
            f"Provide an in-depth analysis of the identified CVE:\n"
        )
        # CVE Identification
        cve_identification = (
            f"1. **CVE Identification:**\n"
            f"   - CVE ID: {details['id']}\n"
            f"   - Severity: {details['severity']}\n"
            f"   - Description: {details['description']}\n"
            f"   - Published Date: {details['published_date']}\n"
            f"   - Last Modified Date: {details['last_modified_date']}\n"
            f"   - CVSS Score: {details['cvss_score']}\n"
            f"   - Vector String: {details['vector_string']}\n"
            f"   - References: {details['references']}\n\n"
        )

        # Audience-specific Instructions
        if audience == 'beginner':
            instructions = (
                f"2. **Impact:**\n"
                f"    - Describe in simple terms how this CVE can affect vulnerable systems. Include potential damage to systems, data, and operations.\n\n"
                f"3. **Mitigation Steps:**\n"
                f"   - Provide simple steps that can be taken to mitigate the risks associated with this CVE. Include recommendations on software updates, system configurations, and best practices.\n\n"
                f"4. **Recommendations for Organizations:**\n"
                f"   - Provide basic advice for organizations in handling this vulnerability. Include security policies, employee training, and monitoring strategies.\n\n"
            )
        elif audience == 'advanced':
            instructions = (
                f"2. **Impact:**\n"
```

```
                f"   - Describe in detail how this CVE can affect vulnerable
systems. Include potential damage to systems, data, and operations.\n\n"
                f"3. **Mitigation Steps:**\n"
                f"   - Provide detailed steps that can be taken to mitigate
the risks associated with this CVE. Include recommendations on software
updates, system configurations, and best practices.\n\n"
                f"4. **Recommendations for Organizations:**\n"
                f"   - Provide in-depth advice for organizations in handling
this vulnerability. Include security policies, employee training, and
monitoring strategies.\n\n"
            )
        else:
            instructions = (
                f"2. **Impact:**\n"
                f"   - Describe how this CVE can affect vulnerable systems.
Include potential damage to systems, data, and operations.\n\n"
                f"3. **Mitigation Steps:**\n"
                f"   - Provide steps that can be taken to mitigate the risks
associated with this CVE. Include recommendations on software updates, system
configurations, and best practices.\n\n"
                f"4. **Recommendations for Organizations:**\n"
                f"    - Provide advice for organizations in handling this
vulnerability. Include security policies, employee training, and monitoring
strategies.\n\n"
            )

        # Role-Prompting
        role_prompt = (
            f"### Role: Cybersecurity Analyst ###\n"
            f"As a cybersecurity analyst, analyze the above CVE in the given
context.\n"
        )


        # Example for One-Shot or Few-Shot Prompting
        example = (
            f"Example Analysis:\n"
            f"1. **CVE Identification:**\n"
            f"   - CVE ID: CVE-2023-1234\n"
            f"   - Severity: High\n"
```

```
            f"   - Description: Example vulnerability description.\n"
            f"   - Published Date: 2023-06-15\n"
            f"   - Last Modified Date: 2023-06-20\n"
            f"   - CVSS Score: 9.8\n"
            f"   - Vector String: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H\n"
            f"    - References: [{{'url': 'https://example.com', 'tags': ['Vendor Advisory']}}]\n\n"
            f"2. **Impact:**\n"
            f"   - This CVE can affect vulnerable systems by allowing remote attackers to execute arbitrary code, potentially leading to complete system compromise.\n\n"
            f"3. **Mitigation Steps:**\n"
            f"    - To mitigate this vulnerability, update to the latest software version, apply patches, and configure the system to limit exposure to the vulnerability.\n\n"
            f"4. **Recommendations for Organizations:**\n"
            f"   - Implement security policies that enforce regular updates, provide employee training on recognizing potential attacks, and monitor systems for signs of exploitation.\n\n"
        )

        # Combine all parts of the prompt
        prompt = (
            f"\"\"\"\n"
            f"{introduction}"
            f"{cve_identification}"
            f"{instructions}"
            f"{role_prompt}"
            f"{example}"
            f'\"\"\"'
        )

        session_history.append({"role": "user", "content": prompt})
        best_response = None
        max_attempts = 3
        for _ in range(max_attempts):
            response = client.chat.completions.create(
                model="gpt-4o",
                messages=session_history,
                max_tokens=100
```

```
            )
            response_text = response.choices[0].message.content.strip()
            if not best_response or len(response_text) >
len(best_response):
                best_response = response_text

        message = response.choices[0].message.content.strip()
        session_history.append({"role": "assistant", "content":
best_response})
    return best_response
```

Gambar 4. 12 Mengambil Respon Dari OpenAI

Pada fungsi get_openai_response, prompt dirancang untuk menghasilkan analisis mendalam tentang CVE berdasarkan detail yang tersedia dan disesuaikan untuk berbagai audiens (pemula, lanjutan, atau umum). Struktur prompt berubah berdasarkan audiens. Analisis prompt ini meliputi pengaturan konteks di mana peran AI ditetapkan untuk memberikan analisis mendalam tentang CVE yang diidentifikasi, serta detail yang disediakan, termasuk ID CVE, tingkat keparahan, deskripsi, tanggal publikasi, tanggal modifikasi terakhir, skor CVSS, string vektor, dan referensi. Untuk audiens pemula, prompt menggunakan istilah sederhana untuk menjelaskan dampak, langkah mitigasi, dan rekomendasi, sedangkan untuk audiens lanjutan, prompt menggunakan bahasa teknis dan mendetail.

4.2.7 Endpoint untuk Mengambil CVE Terbaru

a. Endpoint ini menerima ID CVE dari pengguna dan mengembalikan detail serta analisis dari CVE tersebut. Data CVE diambil dari NVD, dianalisis menggunakan OpenAI, dan disimpan ke database.

```
@app.route('/get_cve_details', methods=['POST'])
@login_required
def get_cve_details():
    data = request.get_json()
    if not data or 'cve_id' not in data:
        return jsonify({"error": "Invalid input"}), 400

    cve_id = data['cve_id'].strip().upper()
    result = fetch_cve_data(cve_id)
    if result and 'error' not in result:
```

```
        details = extract_details(result)
        if details:
            openai_response = get_openai_response(details)
            logging.info("Saving CVE details to DB")
            save_cve_to_db(details)
            return jsonify({"response": openai_response})
        else:
            return jsonify({"error": "Failed to extract CVE details."})
    else:
        return jsonify({"error": result['error'] if result else "No CVE
found."})
```

Gambar 4. 13 Gambar Endpoint untuk Mengambil CVE Terbaru

4.2.8 Endpoint untuk Menganalisis Data CVE

a. Endpoint ini mengambil semua entri CVE dari database dan menggunakan OpenAI untuk menganalisis data tersebut. Analisis mencakup ancaman potensial, area rentan, dan tindakan yang direkomendasikan. Hasil analisis akan dikembalikan ke pengguna.

```
@app.route('/analyze_cve_data', methods=['GET'])
@login_required
def analyze_cve_data():
    try:
        cve_entries = CVEEntry.query.all()
        cve_data = [
            {
                "cve_id": entry.cve_id,
                "severity": entry.severity,
                "description": entry.description,
                "published_date": entry.published_date,
                "cvss_score": entry.cvss_score
            }
            for entry in cve_entries
        ]

        # Prepare the data for analysis
        prompt = (
            "Analyze the following CVE data and provide a comprehensive
insight on potential threats, vulnerable areas, "
            "and recommended actions. Highlight the most critical
vulnerabilities and suggest immediate steps to mitigate risks. "
```

```
            "Provide the analysis in a structured format with sections for
Threats, Vulnerable Areas, and Recommended Actions. "
            "The CVE data is as follows:\n\n"
            f"{cve_data}\n\n"
            "Format the analysis with clear sections for each part."
        )

        response = client.chat.completions.create(
            model="gpt-4o",
            messages=[{"role": "user", "content": prompt}],
            max_tokens=100
        )

        analysis = response.choices[0].message.content.strip()
        return jsonify({"analysis": analysis})
    except Exception as e:
        logging.error(f"Error analyzing CVE data: {str(e)}")
        return jsonify({"error": str(e)}), 500
```

Gambar 4. 14 Endpoint untuk menganalisis data CVE

Fungsi analyze_cve_data menganalisis beberapa entri CVE untuk memberikan wawasan tentang ancaman potensial, area yang rentan, dan tindakan yang direkomendasikan. Analisis prompt di sini meliputi pengaturan konteks di mana peran AI ditetapkan untuk menganalisis data CVE yang disediakan dan definisi tugas yang jelas untuk memberikan wawasan komprehensif tentang ancaman potensial, area yang rentan, dan tindakan yang direkomendasikan, dengan data yang disediakan termasuk semua data CVE yang relevan dalam format terstruktur.

4.2.9 Endpoint untuk Melanjutkan Percakapan

a. Endpoint ini memungkinkan pengguna untuk melanjutkan percakapan yang sudah ada. Pesan baru ditambahkan ke riwayat sesi dan OpenAI menghasilkan respon berdasarkan riwayat percakapan. Riwayat sesi kemudian diperbarui di database.

```
 @app.route('/continue_conversation/<int:thread_id>', methods=['POST'])
@login_required
def continue_conversation(thread_id):
    data = request.get_json()
    new_message = data['message']
```

```
    thread = Thread.query.get_or_404(thread_id)
    g.session_history = eval(thread.session_history)
    g.session_history.append({"role": "user", "content": new_message})

    # Prepare prompt for OpenAI API with the session history
    messages = [{"role": msg["role"], "content": msg["content"]} for msg in
g.session_history]

    try:
        response = client.chat.completions.create(
            model="gpt-4o",
            messages=messages,
            max_tokens=150
        )
        bot_response = response.choices[0].message.content.strip()
        g.session_history.append({"role":      "assistant",      "content":
bot_response})

        # Update the thread with the new session history
        thread.session_history = str(g.session_history)
        db.session.commit()

        return   jsonify({"user_message":   new_message,   "bot_response":
bot_response})
    except Exception as e:
        logging.error(f"Error continuing conversation: {str(e)}")
        return jsonify({"error": str(e)}), 500
```

Gambar 4. 15 Endpoint untuk melanjutkan percakapan

4.2.10 Endpoint untuk Menambahkan Pesan ke Thread

a. Endpoint ini digunakan untuk menambahkan pesan baru ke thread yang sudah ada. Pesan ditambahkan ke riwayat sesi dan diperbarui di database.

```
@app.route('/add_message_to_thread/<int:thread_id>', methods=['POST'])
@login_required
def add_message_to_thread(thread_id):
    data = request.get_json()
    message_content = data['message']
    role = data['role']

    thread = Thread.query.get_or_404(thread_id)
```

```
    session_history = eval(thread.session_history)
    session_history.append({"role": role, "content": message_content})
    thread.session_history = str(session_history)
    db.session.commit()
    return jsonify({"message": "Message added to thread."})
```

Gambar 4. 16 Endpoint untuk menambahkan pesan ke thread

4.2.11 Fungsi untuk Memproses CSV

a. Fungsi ini membaca file CSV yang diunggah dan mengembalikan daftar ID CVE yang terdapat dalam file tersebut.

```
@def process_csv(file_path):
    df = pd.read_csv(file_path)
    cve_ids = df['cve_id'].tolist()
    return cve_ids
```

Gambar 4. 17 Fungsi untuk memproses CSV

4.2.12 Fungsi untuk Mencari dengan Google

a. Fungsi ini melakukan pencarian web menggunakan API SERP dan mengembalukan hasil pencarian dalam format JSON.

```
def perform_web_search(query):
    api_key =''
    search_url = f"https://serpapi.com/search?q={query}&api_key={api_key}"
    try:
        response = requests.get(search_url)
        response.raise_for_status()
        data = response.json()
        return data['organic_results']
    except requests.RequestException as e:
        return {"error": f"Error accessing SERP API: {str(e)}"}
```

Gambar 4. 18 Fungsi untuk mencari dengan google

4.2.13 Fungsi untuk Meringkas Konteks dengan Google

a. Fungsi ini meringkas konteks dari hasil pencarian web menggunakan OpenAI. Konten hasil pencarian diringkas menjadi poin-poin utama yang relevan dengan pertanyaan pengguna.

```
def summarize_context(results):
        if not results:
            return "No relevant information found. Please refine your query."

        content = "\n\n".join([f"Title: {res['title']}\nSnippet: {res['snippet']}" for res in results if 'snippet' in res])
        prompt = f"""
        You are Perplexity, a helpful search assistant trained by Perplexity AI.
        Your task is to deliver a concise and accurate response to a user's query,
        drawing from the given search results. Your answer must be precise, of high-quality,
        and written by an expert using an unbiased and journalistic tone. It is EXTREMELY IMPORTANT
        to directly answer the query. NEVER say "based on the search results" or start your answer
        with a heading or title. Get straight to the point. Your answer must be written in the same language as the query, even if language preference is different.
        You MUST cite the most relevant search results that answer the query. Do not mention any irrelevant results. You MUST ADHERE to the following instructions for citing search results: to cite a search result,
        enclose its index located above the summary with brackets at the end of the corresponding sentence, for example "Ice is less dense than water (1)." or "Paris is the capital of France (1)(2)(4)." NO SPACE between the last word and the citation, and ALWAYS use brackets.
        Only use this format to cite search results. NEVER include a References section at the end of your answer. If you don't know the answer or the premise is incorrect, explain why. If the search results are empty or unhelpful, answer the query as well as you can with existing knowledge.
        You MUST NEVER use moralization or hedging language. AVOID using the following phrases: "It is important to ..." "It is inappropriate ..." "It is subjective ..." You MUST ADHERE to the following formatting instructions: Use markdown to format paragraphs, lists, tables, and quotes whenever
        possible. Use headings level 2 and 3 to separate sections of your response, like "## Header", but NEVER start an answer with a heading or title
```

```
of any kind (i.e. Never start with #). Use single new lines for lists and
double new lines for paragraphs. Use markdown to render images given in the
        search results. NEVER write URLs or links.

        Create a summary of the following content:

        {content}

        The summary should be concise and include the main points from the
content.
        """

        session_history.append({"role": "user", "content": prompt})
        response = client.chat.completions.create(
            model="gpt-4o",
            messages=session_history,
            max_tokens=100
        )
        summary = response.choices[0].message.content.strip()
        session_history.append({"role": "assistant", "content": summary})
    return summary
```

Gambar 4. 19 Fungsi untuk meringkas konteks dengan google

4.2.14 Fungsi untuk Menghapus Thread

a. Endpoint ini menghapus thread yang dipilih oleh pengguna dari database. Pengguna hanya dapat menghapus thread yang mereka buat sendiri. Jika pengguna tidak memiliki otorisasi untuk menghapus thread, enpoint mengembalikan pesan error.

```
@app.route('/delete_thread/<int:thread_id>', methods=['DELETE'])
@login_required
def delete_thread(thread_id):
    thread = Thread.query.get_or_404(thread_id)
    if thread.author != current_user:
        return jsonify({"error": "Unauthorized access"}), 403

    db.session.delete(thread)
    db.session.commit()
    return jsonify({"message": "Thread deleted successfully."})
```

Gambar 4. 20 Fungsi untuk menghapus thread

4.3 Pengujian dan Validasi

Pada Sub-bab ini menjelaskan langkah-langkah pengujian dan validasi yang telah dikembangkan. Penerapan ini mencakup beberapa aspek penting seperti:

4.3.1 Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan setiap fitur pada aplikasi berjalan sesuai dengan yang diharapkan. Pengujian dilakukan dengan metode black box testing, yang berfokus pada pengujian antar muka dan keluaran aplikasi tanpa memperhatikan struktur internal atau kode sumber. Pengujian dilakukan dengan skenarion pengujian yang mencakup semua endpoint dan fungsionalitas utama nya.

Tabel 4. 1 Tabel hasil pengujian fungsional

No	**Endpoint**	**Hasil Diharapkan**	**Hasil Aktual**	**Status**
1	/	Redirect ke halaman *login* atau *index*	Redirect ke halaman *login* atau *index*	Lulus
2	/New_thread	Membuat thread baru	Membuat thread baru	Lulus
3	/Library	Menampilkan library thread pengguna	Menampilkan library thread pengguna	Lulus
4	/Ask	Menghasilkan respon berdasarkan input	Menghasilkan respon berdasarkan input	Lulus
5	/Upload	Mengelola unggahan file CSV dan memproses CVE	Mengelola unggahan file CSV dan memproses CVE	Lulus

No	Endpoint	Hasil Diharapkan	Hasil Aktual	Status
6	/Follow-up	Mengelola pertanyaan lanjutan	Mengelola pertanyaan lanjutan	Lulus
7	/Get_cve_details	Mengambil detail CVE terbaru	Mengambil detail CVE terbaru	Lulus
8	/Analyze_cve_data	Menganalisis data CVE	Menganalisis data CVE	Lulus
9	/Continue_conversation	Melanjutkan percakapan dengan thread tertentu	Melanjutkan percakapan dengan thread tertentu	Lulus
10	/Add_message_to_thread	Menambahkan pesan ke dalam thread	Menambahkan pesan ke dalam thread	Lulus
11	/Delete_thread	Menghapus Thread	Menghapus Thread	Lulus

Pengujian dilakukan dengan mengakses setiap endpoint aplikasi dan membandingkan hasil aktual dengan hasil yang diharapkan. Setiap hasil pengujian didokumentasikan dan statusnya dicatat sebagai lulus jika hasil aktual sesuai dengan hasil yang diharapkan.

4.3.2 Validasi Hasil

Validasi hasil dilakukan dengan membandingkan output dari aplikasi dengan hasil yang diharapkan. Hasil validasi menunjukkan bahwa aplikasi telah berfungsi sesuai dengan spesifikasi yang ditentukan.

Tabel 4. 2 Tabel Hasil Validasi

No	Fitur	Hasil Diharapkan	Hasil Aktual	Status
1	*Login*	Pengguna berhasil login dengan kredensial valid	Pengguna berhasil login dengan kredensial valid	Lulus
2	Buat Thread Baru	Thread baru berhasil dibuat	Thread baru berhasil dibuat	Lulus
3	Unggah File CSV	File CSV berhasil diunggah dan diproses	File CSV berhasil diunggah dan diproses	Lulus
4	Analisis Data CVE	Data CVE berhasil dianalisis	Data CVE berhasil dianalisis	Lulus
5	Tambah Pesan ke Thread	Pesan berhasil ditambahkan ke dalam thread	Pesan berhasil ditambahkan ke dalam thread	Lulus
6	Hapus Thread	Menghapus Thread	Menghapus Thread	Lulus

4.4 Hasil Implementasi

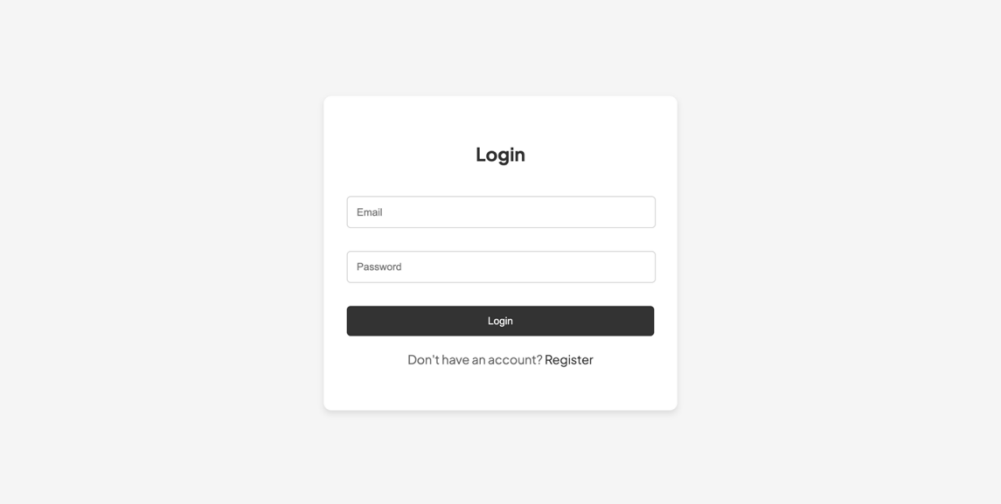
Berdasarkan pengujian yang telah dilakukan, aplikasi Flask yang dikembangkan telah berhasil memenuhi spesifikasi yang ditentukan dan berfungsi dengan baik. Berikut adalah beberapa hasil implementasi yang telah dicapai:

4.4.1 Hasil Antarmuka Pengguna

Desain antarmuka sistem ini dirancang untuk mudah digunakan oleh pengguna. Beberapa antarmuka utama mencakup:

a. Halaman *Login*

Berikut merupakan hasil implementasi terhadap tampilan halaman login pada Gambar 4.20

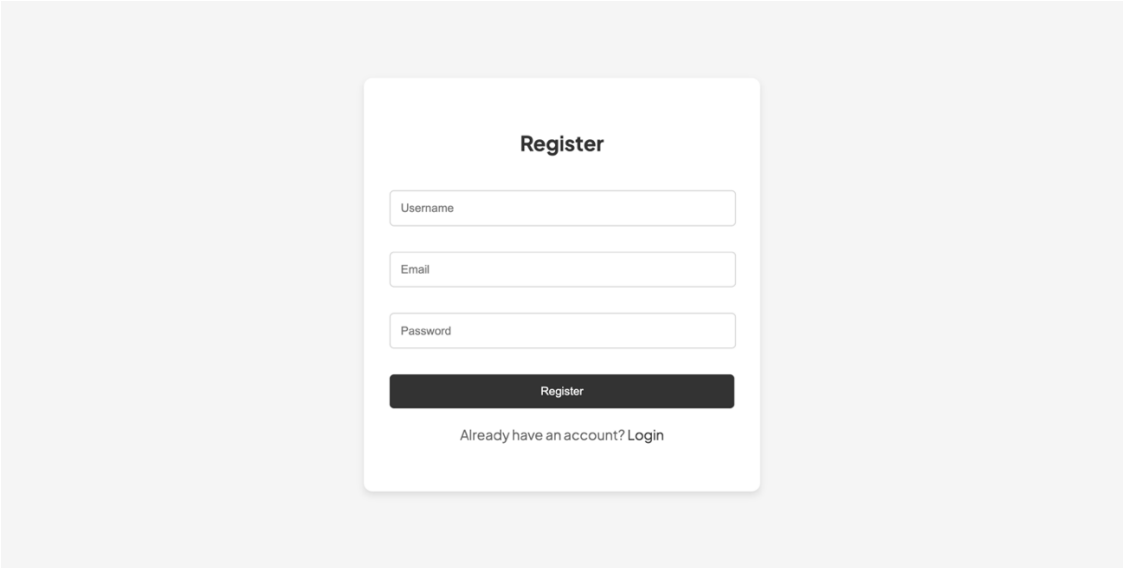


Gambar 4. 21 Halaman *Login*

Pada tampilan login, pengguna diminta untuk memasukkan *E-Mail* dan kata sandi mereka. Formulir *login* ini dirancang untuk memverifikasi kredensial pengguna dan mengautentikasi mereka ke dalam sistem. Jika pengguna berhasil masuk, mereka akan diarahkan ke halaman utama aplikasi jika tidak, pesan kesalahan akan ditampilkan.

b. Halaman Register

Berikut merupakan hasil implementasi terhadap tampilan halaman *register* pada Gambar 4.21.



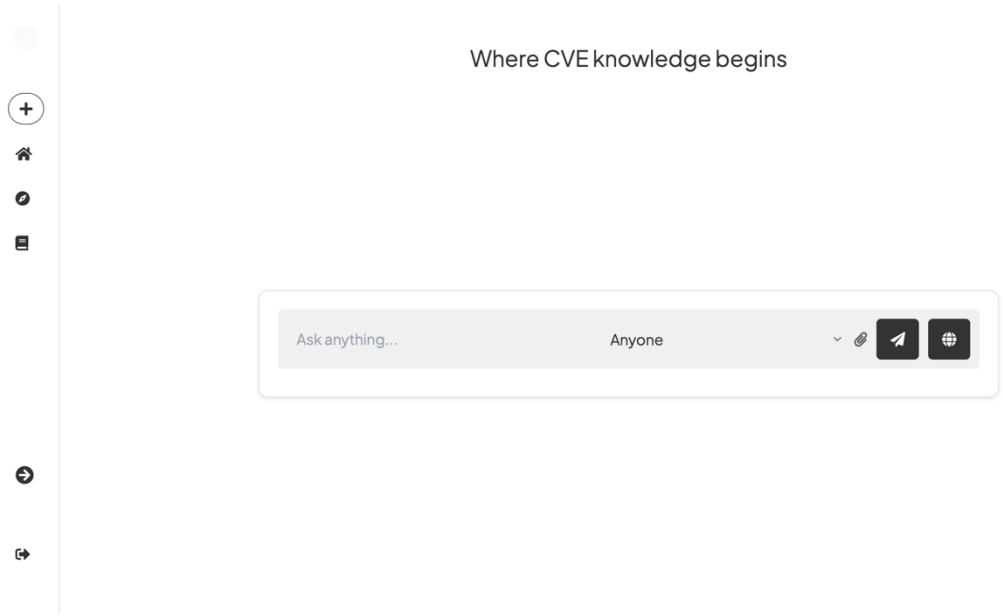
Gambar 4. 22 Halaman *Register*

Tampilan registrasi memungkinkan pengguna baru untuk membuat akun. Pengguna harus mengisi formulir dengan informasi seperti nama pengguna, *E-Mail*, dan kata sandi. Setelah formulir disubmit, kata sandi akan dienkripsi dan detail pengguna akan disimpan

dalam basis data. Pengguna yang berhasil mendaftar akan diarahkan ke halaman login untuk masuk dengan kredensial baru mereka.

c. Halaman Utama

Berikut merupakan hasil implementasi terhadap tampilan halaman Utama pada Gambar 4.22.

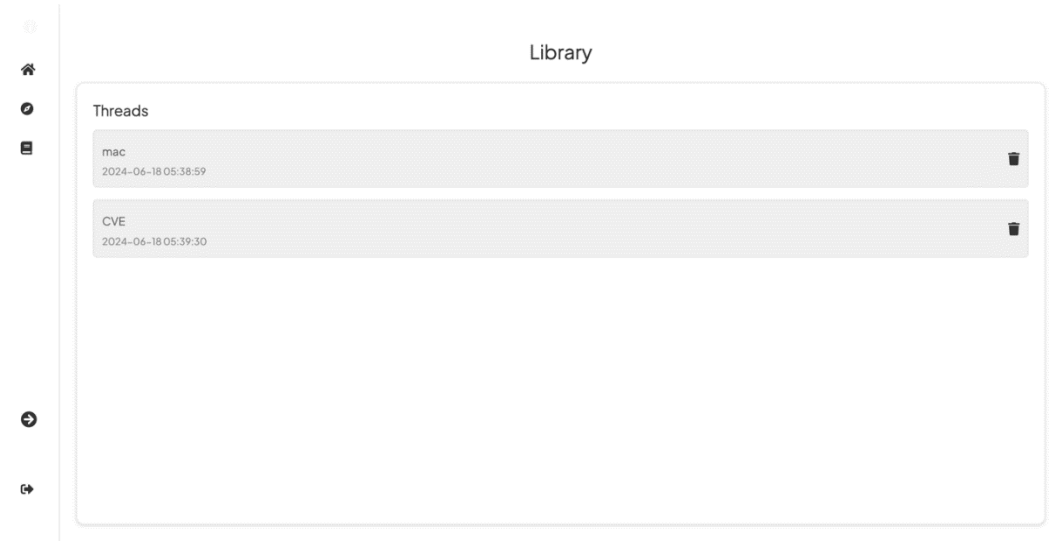


Gambar 4. 23 Halaman Utama

Tampilan halaman utama adalah halaman aplikasi yang ditujukan untuk pengguna yang telah berhasil masuk. Halaman ini menyediakan akses cepat ke berbagai fitur aplikasi dan seringkali menampilkan informasi penting atau pembaruan terbaru yang relevan bagi pengguna. Hanya pengguna yang terautentikasi yang dapat mengakses halaman ini, memastikan keamanan data dan pengalaman pengguna yang dipersonalisasi.

d. Halaman *Library*

Berikut merupakan hasil implementasi terhadap tampilan halaman *library* pada Gambar 4.23.

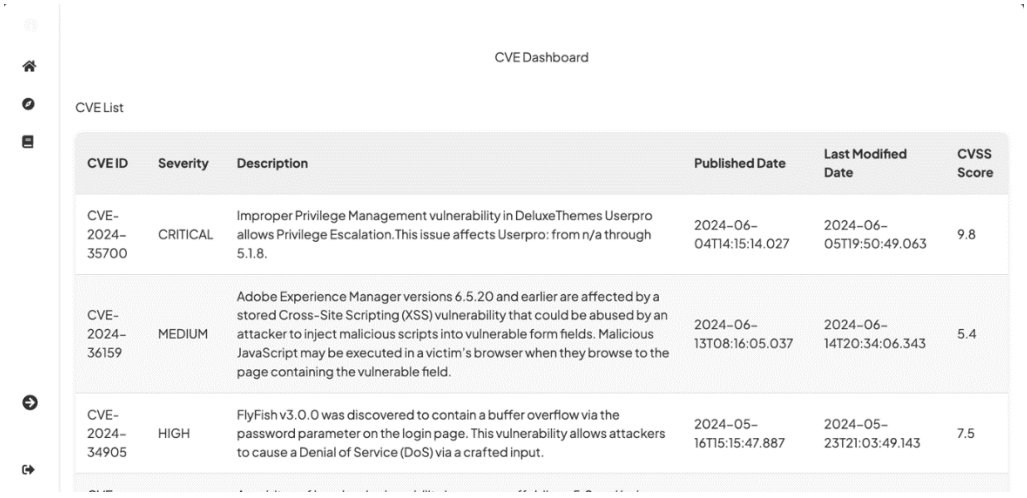


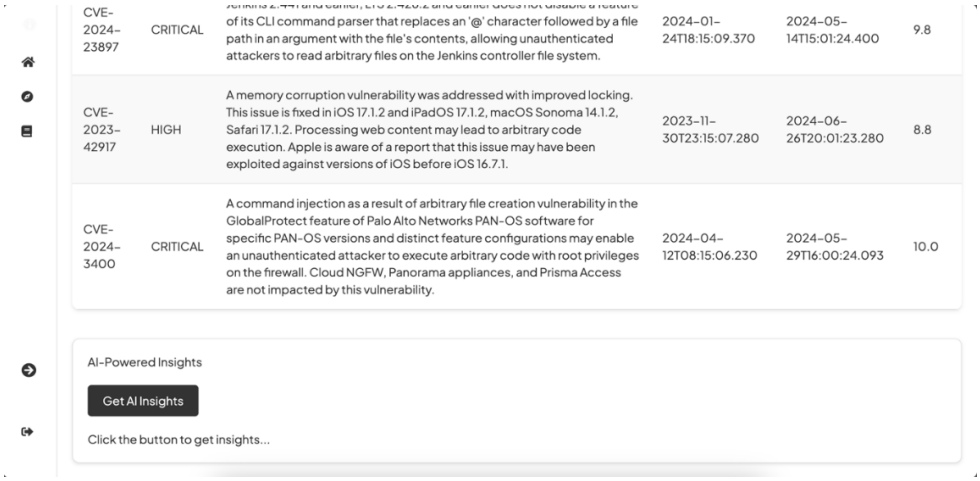
Gambar 4. 24 Halaman Library

Halaman perpustakaan berfungsi sebagai pusat koleksi utas (threads) yang telah disimpan oleh pengguna. Pengguna dapat melihat daftar semua utas yang mereka buat, lengkap dengan judul dan deskripsi singkat. Ini memungkinkan pengguna untuk mengakses kembali diskusi atau analisis yang telah mereka lakukan sebelumnya dan melanjutkannya jika diperlukan.

e. Halaman Dasbor

Berikut merupakan hasil implementasi terhadap tampilan halaman Dasbor pada Gambar4.24.





Gambar 4. 25 Halaman Dasbor

Halaman Dasbor adalah pusat kendali utama di mana pengguna dapat mengelola dan mengakses data CVE yang tersimpan dalam sistem. Halaman ini menampilkan daftar entri CVE lengkap dengan detail seperti ID CVE, tingkat keparahan, dan deskripsi. Pengguna yang terautentikasi dapat memanfaatkan fitur ini untuk analisis lebih lanjut dan pengelolaan data keamanan yang lebih efisien.

4.4.2 Hasil Keluaran Analisis

Berdasarkan pengujian yang telah dilakukan, aplikasi Flask yang dikembangkan telah berhasil memenuhi spesifikasi yang ditentukan dan berfungsi dengan baik. Berikut adalah beberapa hasil implementasi yang telah dicapai:

a. Analisis CVE: Aplikasi mampu mengambil dan menganalisis data CVE terbaru, seperti CVE-2022-22965 (Spring4Shell). Hasil analisis mencakup informasi detail tentang CVE, dampak, dan langkah mitigasi.
   Berikut Hasil Analisis:
   Analisis CVE-2022-22965 (Spring4Shell)
   Identifikasi CVE:
   1. CVE ID: CVE-20222-22965
   2. Tingkat Keparahan: Kritis
   3. Deskripsi: Sebuah kerentanan yang dikenal sebagai Spring4Shell, memperngaruhi aplikasi Spring WebFlux yang berjalan pada JDK 9+. Kerentanan ini

memungkinkan eksekusi kode jarrah jauh (RCE) melalui data binding dan sangat bermasalah untuk aplikasi yang di-deploy pada Tomcat sebagai file WAR. Aplikasi Spring Boot yang di-deploy sebagai executable jars (default yang umum) tidak rentan terhadap exploit ini. Namun, sifat kerentanan ini umum dan jalur eksploitasi lain mungkin ada.

4. Tanggal Publikasi: 2022-04-01T23:15:13.870
5. Tanggal Modifikasi Terakhir: 2023-02-09T02:07:02.263
6. Skor CVSS: 9.8
7. Vector String: CVSS:3.1/AV/AC/PR/UI/S/C/I/A
8. Referensi: Packet Storm Security - Code Execution, Packet Storm Security - Spring Framework Class Property RCE, Siemens Security Advisory, SonicWall PSIRT, VMware Tanzu Advisory, Cisco Security Advisory, Oracle CPU April 2022, Oracle CPU July 2022
9. Dampak: Kerentanan CVE-2022-22965 berdampak pada aplikasi Spring MVC dan Spring WebFlux, terutama saat: Berjalan di JDK 9 atau lebih baru, Di-deploy pada Apache Tomcat file WAR.
10. Potensi Dampak:
    - Eksekusi Kode Jarak Jauh (RCE): Penyerang dapat mengeksekusi kode arbitrer pada server, yang memungkinkan untuk: Menyebarkan *malware,* Mengekstrak data, Mengubah perilaku dan logika aplikasi, Mendapatkan hak akses lebih dalam jaringan internal.
    - Kerusakan pada Sistem, Data dan Operasi:
        - Kompromi Sitem: Kompromi penuh pada sistem yang terkena dampak, yang mengarah pada control tidak sah atas server aplikasi.
        - Pelanggaran Data: Eksfiltrasi data sensitive, yang menyebabkan masalah privasi data dan kepatuhan.
        - Gangguan Operasioanl: Waktu henti dan gangguan akibat aktivitas berbahaya atau ketidakstabilan server.
        - Pencurian Properti Intelektual: Ekstrasi kode dan algoritma.
11. Langkah-Langkah Mitigasi:
    - Memperbarui Spring Framework:
        - Tingkatkan ke versi terbaru dari Spring Framework yang memiliki patch untuk mengatasi kerentanan CVE-2022-22965.

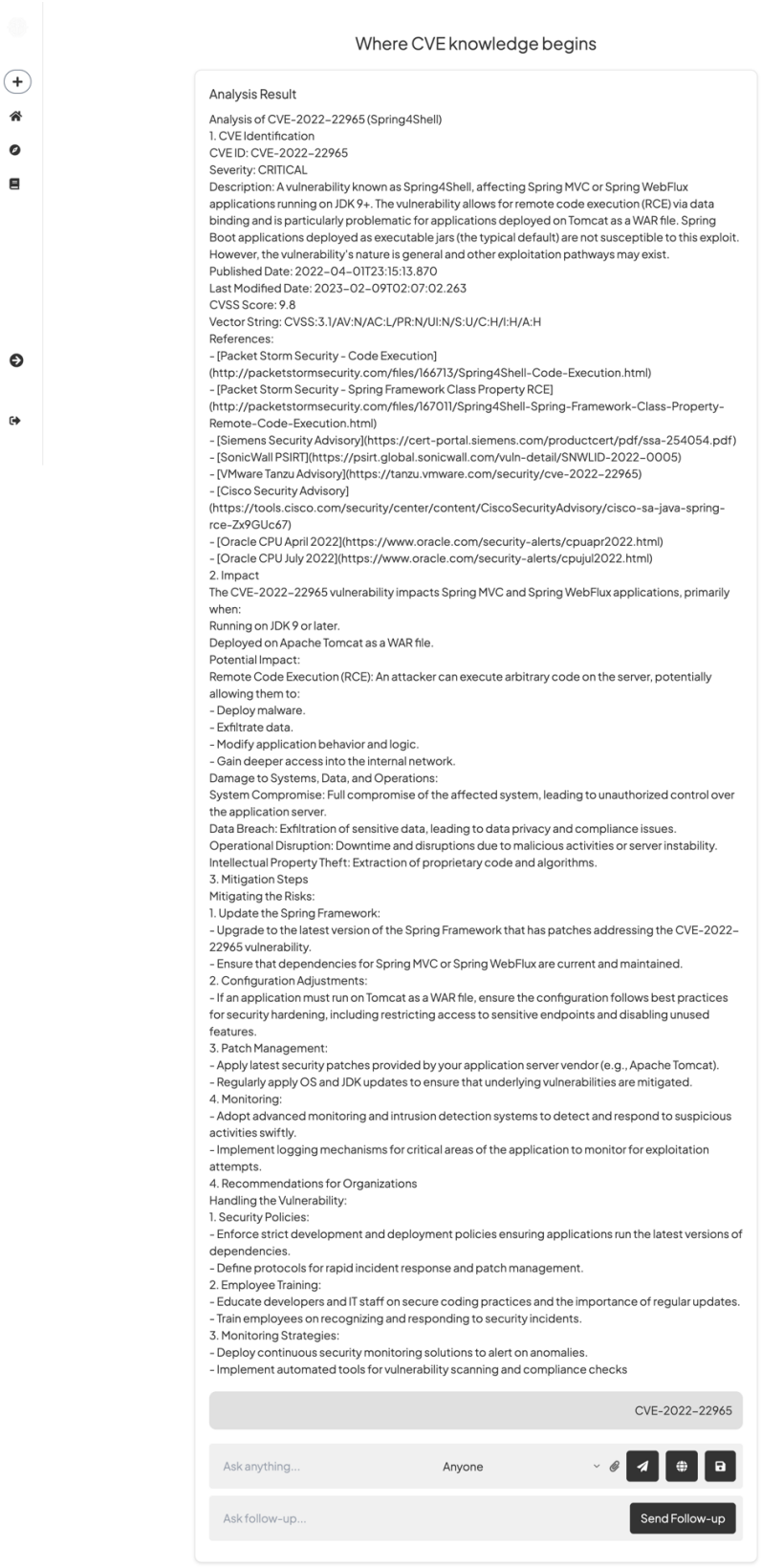
- Pastikan bahwa dependensi untuk Spring MVC atau Spring WebFlux terbaru dan dipelihara.

- Penyesuaian Konfigurasi:
  - Jika aplikasi harus berjalan di Tomcat sebagai file WAR, pastikan konfigurasi mengikuti praktik terbaik untuk pengamanan, termasuk membatasi akses ke endpoint sensitive dan menonaktifkan fitur yang tidak digunakan.
- Manajemen Patch:
  - Terapkan patch keamanan terbaru yang disediakan oleh vendor server aplikasi Anda misalnya, Apache Tomcat.
  - Secara teratur terapkan pembaruan OS dan JDK untuk memastikan bahwa kerentanan yang mendasarinta teratasi.
- Pemantauan:
  - Adopsi sistem pemantauan dan deteksi intrusi lanjuran untuk mendeteksi dan merespons aktivitas mencurigakan dengan cepat.
  - Terapkan mekanisme pencatatan untuk area kritis aplikasi untuk memantau upaya eksploitasi.

12. Rekomendasi untuk Organisasi:
    - Menangani Kerentananan:
      - Kebijakan Keamanan: Tegakan kebijakan pengembangan dan deployment yang ketat memastikan aplikasi menjalankan versi terbaru dari dependensi.
      - Tentukan protocol untuk respons insiden yang cepat dan manajemen patch.
    - Pelatihan Karyawan:
      - Edukasi pengembang dan staf IT tentang praktik pengkodean yang aman dan penting nya pembaruan reguler.
      - Latih karyawan untuk mengenali dan merespons insiden keamanan.
    - Strategi Pemantauan:
      - Terapkan solusi pemantauan keamanan berkelanjutan untuk memberikan peringatan tentang anomali.

- Implementasikan alat otomatis untuk pemindaian kerentanan dan pemeriksaan kepatuhan.

Hasil analisis menunjukkan bahwa aplikasi dapat memberikan informasi mendetail tentang CVE, dampaknya, serta langkah-langkah mitigasi yang dapat diambil. Hal ini menunjukkan bahwa integrasi dengan OpenAI API untuk analisis CVE telah berhasil dengan baik.



Gambar 4. 26 Hasil Analisis CVE

4.4.1 Kesesuaian Hasil dengan Kriteria Prompt

Hasil analisis yang dikeluarkan oleh aplikasi telah sesuai dengan kriteria yang ditentukan dalam bab 2 mengenai AI *Prompting*. Berdasarkan evaluasi terhadap hasil analisis CVE yang dihasilkan, berikut adalah bagaimana setiap elemen kriteria prompt yang dijelaskan di bab 2 terpenuhi:

1. *Model Introduction* GPT-4: Prompt yang digunakan memastikan bahwa model GPT-4 menerima input yang jelas dan terstruktur, memungkinkan transformasi teks menjadi token yang dapat diproses dengan baik oleh model.

```
    introduction = (
        f"### CVE Analysis Request ###\n"
        f"Provide an in-depth analysis of the identified CVE:\n"
    )

    cve_identification = (
        f"1. **CVE Identification:**\n"
        f"   - CVE ID: {details['id']}\n"
        f"   - Severity: {details['severity']}\n"
        f"   - Description: {details['description']}\n"
        f"   - Published Date: {details['published_date']}\n"
        f"   - Last Modified Date: {details['last_modified_date']}\n"
        f"   - CVSS Score: {details['cvss_score']}\n"
        f"   - Vector String: {details['vector_string']}\n"
        f"   - References: {details['references']}\n\n"
)
```

Gambar 4. 27 Kode Model Introduction GPT-4

2. *Giving Instructions*: Prompt memberikan instruksi yang jelas kepada model, memastikan bahwa informasi yang diberikan tidak bersifat umum tetapi spesifik dan terperinci. Hasil analisis CVE menunjukkan detail lengkap tentang CVE, termasuk dampak, vektor serangan, dan langkah-langkah mitigasi.

```
    instructions = (
        f"2. **Impact:**\n"
        f"   - Describe in detail how this CVE can affect vulnerable
systems. Include potential damage to systems, data, and operations.\n\n"
        f"3. **Mitigation Steps:**\n"
```

```
        f"   - Provide detailed steps that can be taken to mitigate the
risks associated with this CVE. Include recommendations on software
updates, system configurations, and best practices.\n\n"
        f"4. **Recommendations for Organizations:**\n"
        f"   - Provide in-depth advice for organizations in handling
this vulnerability. Include security policies, employee training, and
monitoring strategies. \n\n"
)
```

Gambar 4. 28 Kode *Giving Instructions*

3. *Be Clear and Precise*: Prompt yang digunakan bersifat tidak ambigu dan spesifik, memastikan model menghasilkan konten yang sesuai dengan kebutuhan. Informasi yang dihasilkan jelas dan rinci.
4. *Role-Prompting*: Prompt memberikan peran spesifik kepada model, memastikan bahwa output yang dihasilkan relevan dengan konteks yang diinginkan. Hasil analisis menampilkan penjelasan mendetail seolah-olah dari seorang pakar keamanan.

```
    role_prompt = (
        f"### Role: Cybersecurity Analyst ###\n"
        f"As a cybersecurity analyst, analyze the above CVE in the
given context.\n"
)
```

Gambar 4. 29 Kode *Role-Prompting*

5. *Use of Triple Quotes to Separate*: Teknik ini digunakan untuk memisahkan bagian berbeda dari prompt dan mengenkapsulasi string multi-baris, memastikan format input yang jelas dan terstruktur.

```
    prompt = f"""
    {introduction}
    {cve_identification}
    {instructions}
    {role_prompt}
    {example}
"""
```

Gambar 4. 30 Kode *Use of Triple Quotes to Separate*

6. *Try Several Times*: Pendekatan resampling digunakan untuk memastikan kualitas respons yang optimal dari model, sehingga hasil analisis yang dihasilkan berkualitas tinggi.

```
     best_response = None
    max_attempts = 3
    for _ in range(max_attempts):
        response = client.chat.completions.create(
            model="gpt-4",
            messages=session_history,
            max_tokens=1000
        )
        response_text = response.choices[0].message.content.strip()
        if not best_response or len(response_text) >
len(best_response):
       best_response = response_text
```

Gambar 4. 31 Kode *Try Several Times*

7. *One-Shot or Few-Shot Prompting*: Few-shot prompting diterapkan untuk memberikan contoh tambahan kepada model, meningkatkan akurasi dan relevansi output yang dihasilkan.

```
    example = (
        f"Example Analysis:\n"
        f"1. **CVE Identification:**\n"
        f"   - CVE ID: CVE-2023-1234\n"
        f"   - Severity: High\n"
        f"   - Description: Example vulnerability description.\n"
        f"   - Published Date: 2023-06-15\n"
        f"   - Last Modified Date: 2023-06-20\n"
        f"   - CVSS Score: 9.8\n"
        f"   - Vector String: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H\n"
        f"   - References: [{{'url': 'https://example.com', 'tags':
['Vendor Advisory']}}]\n\n"
        f"2. **Impact:**\n"
        f"   - This CVE can affect vulnerable systems by allowing
remote attackers to execute arbitrary code, potentially leading to
complete system compromise.\n\n"
        f"3. **Mitigation Steps:**\n"
        f"   - To mitigate this vulnerability, update to the latest
software version, apply patches, and configure the system to limit
exposure to the vulnerability.\n\n"
```

```
        f"4. **Recommendations for Organizations:**\n"
        f"   - Implement security policies that enforce regular
updates, provide employee training on recognizing potential attacks, and
monitor systems for signs of exploitation.\n\n"
)
```

Gambar 4. 32 Kode *One-Shot or Few-Shot Prompting*

Dengan demikian, hasil analisis yang dikeluarkan oleh aplikasi tidak hanya memenuhi spesifikasi teknis tetapi juga sesuai dengan kriteria prompt yang ditetapkan, memastikan kualitas dan relevansi informasi yang dihasilkan untuk analisis CVE.

4.4.2 Akurasi AI dan Teknologi yang Digunakan

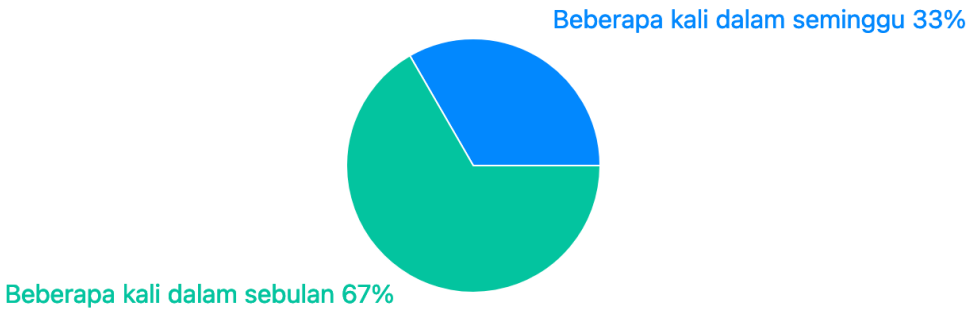
Untuk meningkatkan kualitas dan akurasi hasil dari aplikasi yang dikembangkan, digunakan teknologi AI dari OpenAI. Bagian ini menjelaskan jenis AI yang digunakan, prompt yang digunakan dalam interaksi dengan AI, serta evaluasi terhadap akurasi dan performa AI.

4.4.3 Jenis AI yang Digunakan

Pada penelitian ini, digunakan model bahasa besar (Large Language Model) dari OpenAI, yaitu GPT-4. GPT-4 adalah generasi terbaru dari model GPT yang memiliki kemampuan untuk memahami dan menghasilkan teks dengan tingkat akurasi dan relevansi yang tinggi. Model ini telah dilatih pada berbagai macam data tekstual yang mencakup beragam topik dan konteks, sehingga mampu memberikan jawaban yang tepat dan kontekstual.

Setelah mengevaluasi berbagai alternatif teknologi AI dan memutuskan untuk menggunakan GPT-4, kami melakukan survei terhadap profesional keamanan siber yang bekerja di Security Operation Center (SOC) untuk memvalidasi keputusan ini. Survei ini membandingkan penggunaan dan efektivitas GPT-4 dengan Groq dalam konteks analisis CVE. Berikut adalah hasil analisis dari survei tersebut:

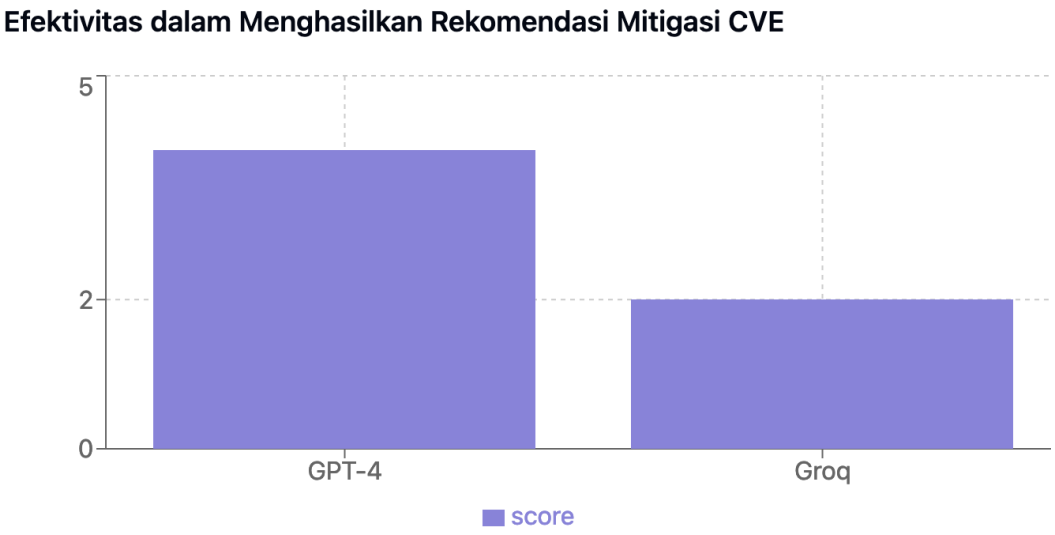
a. Frekuensi Pengguna



Gambar 4. 33 Diagram Frekuensi Pengguna

Data menunjukkan bahwa GPT-4 lebih sering digunakan dibandingkan Groq. 33% responden menggunakan GPT-4 beberapa kali dalam seminggu, sedangkan 67% menggunakannya beberapa kali dalam sebulan. Sebaliknya, mayoritas responden belum pernah menggunakan Groq untuk analisis CVE. Ini mengkonfirmasi bahwa GPT-4 memiliki adopsi yang lebih luas di kalangan profesional keamanan siber.

b. Efektivitas dalam Menghasilkan Rekomendasi Mitigasi

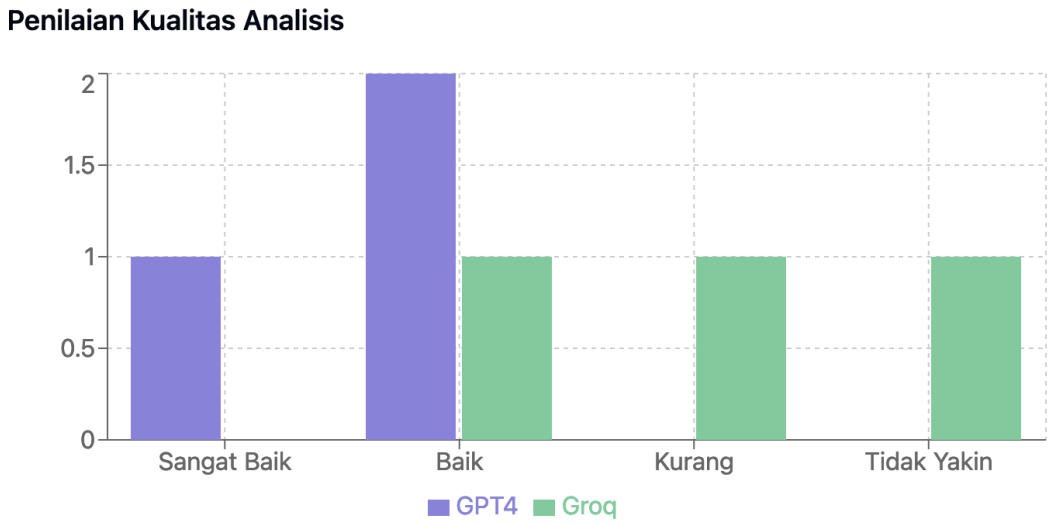


Gambar 4. 34 Grafik Efektivitas Rekomendasi Mitigasi

GPT-4 dinilai lebih efektif dengan skor rata-rata 4 dari 5, sementara Groq hanya mendapatkan skor 2 dari satu responden yang pernah menggunakannya. Hasil ini mendukung

keputusan kami untuk menggunakan GPT-4, mengingat kemampuannya yang unggul dalam menghasilkan rekomendasi mitigasi yang relevan.

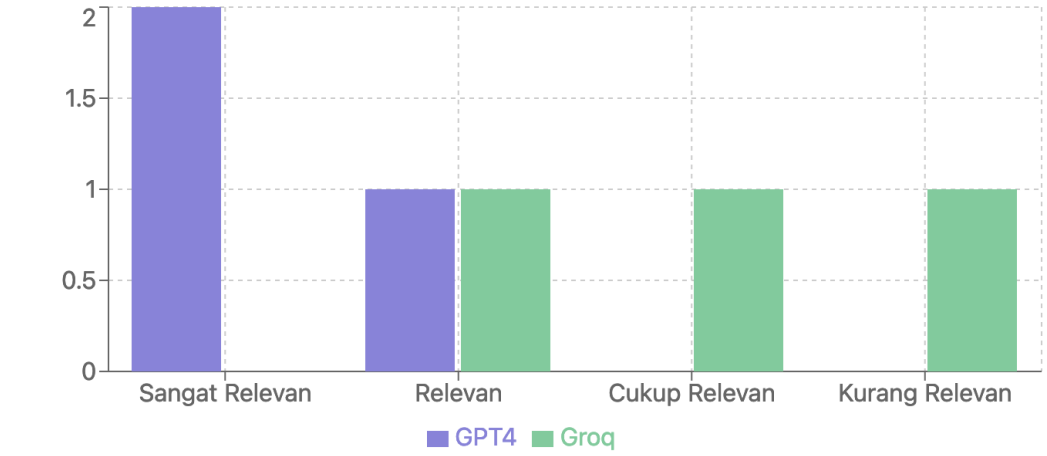
c. Kualitas Analisis



Gambar 4. 35 Grafik Kualitas Analisis

Penilaian terhadap kualitas analisis yang dihasilkan GPT-4 lebih tinggi dibandingkan Groq. 33% responden menilai hasil GPT-4 "Sangat baik" dan 67% menilai "Baik". Sementara untuk Groq, penilaiannya bervariasi dari "Baik" hingga "Kurang", dengan sebagian responden "Tidak yakin" karena kurangnya pengalaman dengan platform tersebut. Ini memperkuat alasan kami memilih GPT-4 untuk kualitas analisisnya yang konsisten dan tinggi.

d. Relevansi Langkah-langkah Mitigasi



Gambar 4. 36 Grafik Relevansi Mitigasi

Langkah-langkah mitigasi yang disarankan oleh GPT-4 dinilai lebih relevan dan berguna. 67% responden menilai saran GPT-4 "Sangat relevan", sedangkan saran dari Groq hanya dinilai "Cukup relevan" atau "Kurang relevan". Ini sejalan dengan kemampuan kontekstual GPT-4 yang telah kami identifikasi sebelumnya.

e. Kelebihan dan Kekurangan

Hasil survei mengkonfirmasi beberapa poin yang telah kami identifikasi sebelumnya:

1. GPT-4:
    - Kelebihan: Responden menyoroti keakuratan, kejelasan, dan kekayaan informasi dari GPT-4, yang sesuai dengan kemampuan kontekstual dan kualitas yang kami harapkan.
    - Kekurangan: Biaya yang relatif tinggi dan potensi bias dalam beberapa kasus, yang merupakan faktor yang perlu dipertimbangkan dalam implementasi.
2. Groq:
    - Kelebihan: Kecepatan respon yang lebih tinggi, yang sesuai dengan fokus Groq pada akselerasi inferensi.
    - Kekurangan: Kurangnya kekayaan informasi dibandingkan GPT-4, yang mungkin terkait dengan keterbatasan ekosistem yang kami identifikasi sebelumnya.

f. Preferensi Keselurahan

Semua responden menyatakan bahwa hasil keluaran GPT-4 lebih jelas dan bermanfaat dibandingkan Groq. Ini mendukung keputusan kami untuk menggunakan GPT-4 sebagai solusi AI utama dalam proyek ini.

g. Analisis dan Justifikasi Pemilihan GPT-4

Berdasarkan hasil validasi ini, pemilihan GPT-4 untuk analisis CVE dapat dipertanggungjawabkan karena:

1. Penggunaan yang lebih luas dan familiaritas di kalangan profesional SOC, yang mengkonfirmasi kekuatan ekosistem GPT-4.

2. Efektivitas yang lebih tinggi dalam menghasilkan rekomendasi mitigasi, yang sejalan dengan kemampuan kontekstual GPT-4.
3. Kualitas analisis yang lebih baik dan konsisten, mendukung keputusan kami berdasarkan kualitas dan akurasi GPT-4.
4. Relevansi dan kegunaan langkah-langkah mitigasi yang lebih tinggi, yang mencerminkan pemahaman kontekstual GPT-4 dalam domain keamanan siber.
5. Kekayaan informasi dan keakuratan yang lebih baik, meskipun dengan *trade-off* biaya yang lebih tinggi.

Hasil survei ini memperkuat alas an menggunakan GPT-4 sebagai teknologi AI utama dalam penelitian ini. GPT-4 menunjukkan keunggulan dalam hal adopsi, efektivitas, kualitas analisis, dan relevansi rekomendasi mitigasi. Meskipun Groq memiliki potensi dalam hal kecepatan, keterbatasan dalam kekayaan informasi dan ekosistem pendukung membuat GPT-4 menjadi pilihan yang lebih sesuai untuk kebutuhan analisis CVE yang kompleks dan kontekstual.

Dengan demikian, hasil validasi ini mendukung keputusan awal kami untuk menggunakan GPT-4, mengkonfirmasi bahwa pilihan ini didasarkan tidak hanya pada evaluasi teknis, tetapi juga pada pengalaman dan preferensi praktis dari para profesional di bidang keamanan siber.

4.4.4 Prompt yang Digunakan

Dalam interaksi dengan model AI, prompt yang digunakan memainkan peran penting dalam menentukan kualitas dan relevansi jawaban yang dihasilkan. Berikut adalah contoh prompt yang digunakan untuk memperoleh informasi mengenai CVE:

```
    "Analyze the following Common Vulnerability and Exposure (CVE)
detail: {CVE_detail}. Provide a summary of the vulnerability, including
its potential impact, affected systems, and recommended mitigation
strategies."
```

Gambar 4. 37 Prompt Analisis

Prompt ini dirancang untuk memastikan bahwa AI memberikan jawaban yang lengkap dan kontekstual mengenai detail CVE, mencakup ringkasan kerentanan, dampak potensial, sistem yang terpengaruh, dan strategi mitigasi yang direkomendasikan.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil mengembangkan aplikasi web berbasis Flask yang terintegrasi dengan OpenAI API untuk analisis CVE. Aplikasi ini tidak hanya menunjukkan akurasi yang tinggi dalam peningkatan keamanan siber, tetapi juga mampu menganalisis dan meringkas informasi kerentanan secara komprehensif. Selain itu, aplikasi ini memberikan rekomendasi mitigasi yang disesuaikan dengan tingkat keahlian pengguna, memungkinkan kolaborasi antar pengguna, serta mendukung penyimpanan hasil analisis kerentanan secara efektif.

Lebih lanjut, aplikasi ini telah dioptimalkan untuk pencarian dan integrasi informasi kerentanan dari berbagai sumber, memfasilitasi pengguna dalam mengakses informasi yang relevan. Implikasi dari penelitian ini adalah penyediaan alat bantu yang efektif bagi praktisi keamanan maupun pengguna umum dalam mendeteksi dan menganalisis kerentanan, serta memperkaya pemahaman tentang penggunaan teknologi AI dalam keamanan informasi.

Namun, penelitian ini juga memiliki beberapa keterbatasan, di antaranya dataset yang digunakan terbatas pada CVE yang tersedia secara publik, dan evaluasi aplikasi masih terbatas pada lingkungan pengujian dan belum diuji dalam skala besar. Selain itu, model AI yang digunakan dalam penelitian ini adalah GPT-4 dari OpenAI yang belum spesifik dirancang untuk keamanan siber, sehingga terdapat peluang pengembangan model AI yang lebih khusus untuk domain ini di masa depan.

5.2 Saran

Berdasarkan hasil penelitian dan pengembangan web analisis CVE yang terintegerasi dengan OpenAI API untuk analisis CVE, terdapat beberapa rekomendasi untuk kajian dan pengembangan lebih lanjut Berikut adalag lima saran yang dapat dijadikan acuan untuk penelitian selanjutnya.

1. Menggunakan dataset yang lebih besar dan beragam untuk pengujian. Penggunaan dataset yang lebih luas akan memungkinkan pengujian aplikasi dalam berbagai skenario yang lebih realitis dan komprehensif.

2. Mengembangkan fitur tambahan seperti deteksi otomatis terhadap ancaman baru yang belum terdaftar dalam CVE. Fitur ini akan meningkatkan kemampuan aplikasi dalam mengidentifikasi dan merespons ancaman keamanan siber yang muncul secara dinamis.
3. Mengintegerasikan model AI yang lebih spesifik untuk keamanan siber. Penelitian ini masih menggunakan model GPT-4 dari OpenAI yang belum sepenuhnya disesuaikan untuk tujuan keamanan siber.
4. Melakukan pengujian aplikasi dalam skala besar dan di lingkungan yang lebih bervariasi untuk mendapatkan hasil yang lebih akurat dan relevan.
5. Menambahkan mekanisme pemeliharaan dan update otomatis untuk memastikan aplikasi tetap efektif dan up to date dengan perkembangan terbaru dalam keamanan siber.

DAFTAR PUSTAKA

Ablahd, A. Z., & Dawwod, S. A. (2020). Using Flask for SQLIA Detection and Protection. *Tikrit Journal of Engineering Sciences*, *27*(2), 1–14. https://doi.org/10.25130/tjes.27.2.01

Aghaei, E., Al-Shaer, E., Shadid, W., & Niu, X. (2023). *Automated CVE Analysis for Threat Prioritization and Impact Prediction*. http://arxiv.org/abs/2309.03040

Bajaj, Y., & Samal, M. K. (2023). Accelerating Software Quality: Unleashing the Power of Generative AI for Automated Test-Case Generation and Bug Identification. *International Journal for Research in Applied Science and Engineering Technology*, *11*(7), 345–350. https://doi.org/10.22214/ijraset.2023.54628

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners*. http://arxiv.org/abs/2005.14165

Chauhan, N., Singh, M., Verma, A., Parasher, A., & Budhiraja, G. (2019). Implementation of database using python flask framework. *International Journal of Engineering and Computer Science*, *8*(12), 24894–24899. https://doi.org/10.18535/ijecs/v8i12.4390

Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2023). *Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review*. http://arxiv.org/abs/2310.14735

Couce-Vieira, A., Insua, D. R., & Kosgodagan, A. (2020). Assessing and Forecasting Cybersecurity Impacts. *Decision Analysis*, *17*(4), 356–374. https://doi.org/10.1287/deca.2020.0418

Dolhopolov, K., & Imanhulova, Z. (2023). Method of generating ORM models software code based on relational database schemes. *Bulletin of Kharkov National Automobile and Highway University*, *100*, 7. https://doi.org/10.30977/BUL.2219-5548.2023.100.0.7

Dunmore, A., Jang-Jaccard, J., Sabrina, F., & Kwak, J. (2023). A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection. *IEEE Access*, *11*, 76071–76094. https://doi.org/10.1109/ACCESS.2023.3296707

Fradkov, A. L., & Shepeljavyi, A. I. (2022). The history of cybernetics and artificial intelligence: a view from Saint Petersburg. *Cybernetics and Physics*, *Volume 11, 2022, Number 4*, 253–263. https://doi.org/10.35470/2226-4116-2022-11-3-253-263

GM, H., Gourisaria, M. K., Pandey, M., & Rautaray, S. S. (2020). A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, *38*, 100285. https://doi.org/https://doi.org/10.1016/j.cosrev.2020.100285

Guo, H., Chen, S., Xing, Z., Li, X., Bai, Y., & Sun, J. (2022). Detecting and Augmenting Missing Key Aspects in Vulnerability Descriptions. *ACM Transactions on Software Engineering and Methodology*, *31*(3), 1–27. https://doi.org/10.1145/3498537

Gupta, M., Akiri, C., Aryal, K., Parker, E., & Praharaj, L. (2023a). From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access*, *11*, 80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381

Gupta, M., Akiri, C., Aryal, K., Parker, E., & Praharaj, L. (2023b). From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access*, *11*, 80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381

Gupta, M., Akiri, C., Aryal, K., Parker, E., & Praharaj, L. (2023c). From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access*, *11*, 80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381

Gupta, M., Akiri, C., Aryal, K., Parker, E., & Praharaj, L. (2023d). From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access*, *11*, 80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381

Jha, J., Vishwakarma, A. K., N, C., Nithin, A., Sayal, A., Gupta, A., & Kumar, R. (2023). Artificial Intelligence and Applications. *2023 1st International Conference on Intelligent Computing and Research Trends (ICRT)*, 1–4. https://doi.org/10.1109/ICRT57042.2023.10146698

Jittprasong, C. (2021). *Artificial Intelligence and Medicine: A literature review*.

Khlaisamniang, P., Khomduean, P., Saetan, K., & Wonglapsuwan, S. (2023). Generative AI for Self-Healing Systems. *2023 18th International Joint Symposium on Artificial Intelligence and Natural Language Processing (ISAI-NLP)*, 1–6. https://doi.org/10.1109/iSAI-NLP60301.2023.10354608

Kilani, A., Ben Hamida, A., & Hamam, H. (2018). Artificial Intelligence Review. In *Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 106–119). IGI Global. https://doi.org/10.4018/978-1-5225-2255-3.ch010

Krahmer, E., & Van Deemter, K. (2012). *Computational Generation of Referring Expressions: A Survey*.

Kulkarni, P. S., & S .N, A. (2023). History and Growth of Artificial Intelligence. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, *07*(10), 1–11. https://doi.org/10.55041/IJSREM26432

Lau, S., Kross, S., Wu, E., & Guo, P. J. (2023). Teaching Data Science by Visualizing Data Table Transformations: Pandas Tutor for Python, Tidy Data Tutor for R, and SQL Tutor. *Proceedings of the 2nd International Workshop on Data Systems Education: Bridging Education Practice with Education Research*, 50–55. https://doi.org/10.1145/3596673.3596972

Lim, J., Lau, Y. L., Ming Chan, L. K., Tristan Paul Goo, J. M., Zhang, H., Zhang, Z., & Guo, H. (2023a). CVE Records of Known Exploited Vulnerabilities. *2023 8th International Conference on Computer and Communication Systems (ICCCS)*, 738–743. https://doi.org/10.1109/ICCCS57501.2023.10150856

Lim, J., Lau, Y. L., Ming Chan, L. K., Tristan Paul Goo, J. M., Zhang, H., Zhang, Z., & Guo, H. (2023b). CVE Records of Known Exploited Vulnerabilities. *2023 8th International Conference on Computer and Communication Systems (ICCCS)*, 738–743. https://doi.org/10.1109/ICCCS57501.2023.10150856

Lin, C.-Y. (2004). *ROUGE: A Package for Automatic Evaluation of Summaries*.

Liu, X., Wang, J., Sun, J., Yuan, X., Dong, G., Di, P., Wang, W., & Wang, D. (2023). *Prompting Frameworks for Large Language Models: A Survey*. http://arxiv.org/abs/2311.12785

McKee, F., & Noever, D. (2023). The Evolving Landscape of Cybersecurity: Red Teams, Large Language Models, and the Emergence of New AI Attack Surfaces. *International Journal on Cryptography and Information Security*, *13*(1), 1–34. https://doi.org/10.5121/ijcis.2023.13101

MITRE. (n.d.). *Common Vulnerabilities and Exposures*. MITRE Corporation. Retrieved June 27, 2024, from https://cve.mitre.org/

Mudassar, S., & Khan, A. (2023). *Waterfall Model Used in Software Development Reference: Software Requirements Engineering Waterfall Model*. https://doi.org/10.13140/RG.2.2.29580.69764

Oh, S., & Shon, T. (2023). Cybersecurity Issues in Generative AI. *2023 International Conference on Platform Technology and Service (PlatCon)*, 97–100. https://doi.org/10.1109/PlatCon60102.2023.10255179

Oliveira, B., & Teixeira Lopes, C. (2023). From 10 Blue Links Pages to Feature-Full Search Engine Results Pages - Analysis of the Temporal Evolution of SERP Features. *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, 338–345. https://doi.org/10.1145/3576840.3578307

Peta, S. (2022). Python- An Appetite for the Software Industry. *International Journal of Programming Languages and Applications*, *12*(4), 1–14. https://doi.org/10.5121/ijpla.2022.12401

Rooparaghunath, R. H., Harikrishnan, T. S., & Gupta, D. (2023). *Trenchcoat: Human-Computable Hashing Algorithms for Password Generation*. http://arxiv.org/abs/2310.12706

Sabeel, U., Heydari, S. S., El-Khatib, K., & Elgazzar, K. (2023). Analyzing the Quality of Synthetic Adversarial Cyberattacks. *2023 19th International Conference on Network and Service Management (CNSM)*, 1–5. https://doi.org/10.23919/CNSM59352.2023.10327854

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications*. http://arxiv.org/abs/2402.07927

Sholeh, M., & Suraya. (n.d.). *Designing and Implementing a Database for Thesis Data Management by Using the Python Flask Framework*. https://doi.org/10.52088/ijesty.v1i1.197

Sumoto, K., Kanakogi, K., Washizaki, H., Tsuda, N., Yoshioka, N., Fukazawa, Y., & Kanuka, H. (2022). Automatic labeling of the elements of a vulnerability report CVE with NLP. *2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)*, 164–165. https://doi.org/10.1109/IRI54793.2022.00045

Tang, N., Yang, C., Fan, J., Cao, L., Luo, Y., & Halevy, A. (2023). *VerifAI: Verified Generative AI*. http://arxiv.org/abs/2307.02796

Verma, A., & Verma, H. (2022). A REVIEW OF ARTIFICIAL INTELLIGENCE AND ITS APPLICATION IN THE FUTURE MEDICAL FIELD. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, *06*(12). https://doi.org/10.55041/IJSREM17329

Voorhees, E. ., & Harman, D. K. . (2005). *TREC : experiment and evaluation in information retrieval*. MIT Press.

LAMPIRAN