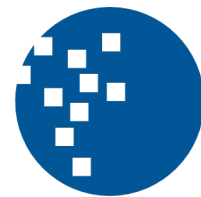


**PENGEMBANGAN ALAT BANTU BERBASIS AI AGENT
UNTUK DETEKSI KERENTANAN BROKEN ACCESS
CONTROL MELALUI OTOMATISASI PENGUJIAN
ENDPOINT PADA APLIKASI BACK-END**



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

SKRIPSI

CAROLINE SUSANTO
00000071280

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2025

**PENGEMBANGAN ALAT BANTU BERBASIS AI AGENT
UNTUK DETEKSI KERENTANAN BROKEN ACCESS
CONTROL MELALUI OTOMATISASI PENGUJIAN
ENDPOINT PADA APLIKASI BACK-END**



Diajukan sebagai salah satu syarat untuk memperoleh
Gelar Sarjana Komputer (S.Kom.)

CAROLINE SUSANTO
00000071280

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2025

HALAMAN PERNYATAAN TIDAK PLAGIAT

Dengan ini saya,

Nama : Caroline Susanto

Nomor Induk Mahasiswa : 00000071280

Program Studi : Informatika

Skripsi dengan judul:

Pengembangan Alat Bantu Berbasis AI Agent untuk Deteksi Kerentanan Broken Access Control melalui Otomatisasi Pengujian Endpoint pada Aplikasi Back-End

merupakan hasil karya saya sendiri bukan plagiat dari laporan karya tulis ilmiah yang ditulis oleh orang lain, dan semua sumber, baik yang dikutip maupun dirujuk, telah saya nyatakan dengan benar serta dicantumkan di Daftar Pustaka.

Jika di kemudian hari terbukti ditemukan kecurangan/penyimpangan, baik dalam pelaksanaan maupun dalam penulisan laporan karya tulis ilmiah, saya bersedia menerima konsekuensi dinyatakan TIDAK LULUS untuk mata kuliah yang telah saya tempuh.

Tangerang, 13 November 2025



(Caroline Susanto)

HALAMAN PENGESAHAN

Skripsi dengan judul
**PENGEMBANGAN ALAT BANTU BERBASIS AI AGENT UNTUK
DETEKSI KERENTANAN BROKEN ACCESS CONTROL
MELALUI OTOMATISASI PENGUJIAN ENDPOINT
PADA APLIKASI BACK-END**

oleh

Nama : Caroline Susanto
NIM : 00000071280
Program Studi : Informatika
Fakultas : Fakultas Teknik dan Informatika

Telah diujikan pada hari Jumat, 21 November 2025
Pukul 15.00 s/s 17.00 dan dinyatakan
LULUS

Dengan susunan penguji sebagai berikut

Ketua Sidang

Penguji

(Dr. Maria Irmina Prasetyowati, S.Kom.,
M.T.)

NIDN: 0725057201

(David Agustriawan, S.Kom., M.Sc.,
Ph.D.)

NIDN: 0525088601

Pembimbing

(Moeljono Widjaja, B.Sc., M.Sc., Ph.D)

NIDN: 0311106903

Ketua Program Studi Informatika,

(Arya Wicaksana, S.Kom., M.Eng.Sc., OCA, CEH, CEI)

NIDN: 0315109103

**HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK
KEPENTINGAN AKADEMIS**

Yang bertanda tangan di bawah ini:

Nama : Caroline Susanto
NIM : 00000071280
Program Studi : Informatika
Jenjang : S1
Judul Karya Ilmiah : Pengembangan Alat Bantu Berbasis
AI Agent untuk Deteksi Kerentanan
Broken Access Control melalui
Otomatisasi Pengujian Endpoint pada
Aplikasi Back-End

Menyatakan dengan sesungguhnya bahwa saya bersedia:

- ☒ Saya bersedia memberikan izin sepenuhnya kepada Universitas Multimedia Nusantara untuk mempublikasikan hasil karya ilmiah saya ke dalam repositori Knowledge Center sehingga dapat diakses oleh Sivitas Akademika UMN/Publik. Saya menyatakan bahwa karya ilmiah yang saya buat tidak mengandung data yang bersifat konfidensial.
- ☐ Saya tidak bersedia mempublikasikan hasil karya ilmiah ini ke dalam repositori Knowledge Center, dikarenakan: dalam proses pengajuan publikasi ke jurnal/konferensi nasional/internasional (dibuktikan dengan *letter of acceptance*) **.
- ☐ Lainnya, pilih salah satu:
- Hanya dapat diakses secara internal Universitas Multimedia Nusantara
 - Embargo publikasi karya ilmiah dalam kurun waktu tiga tahun.

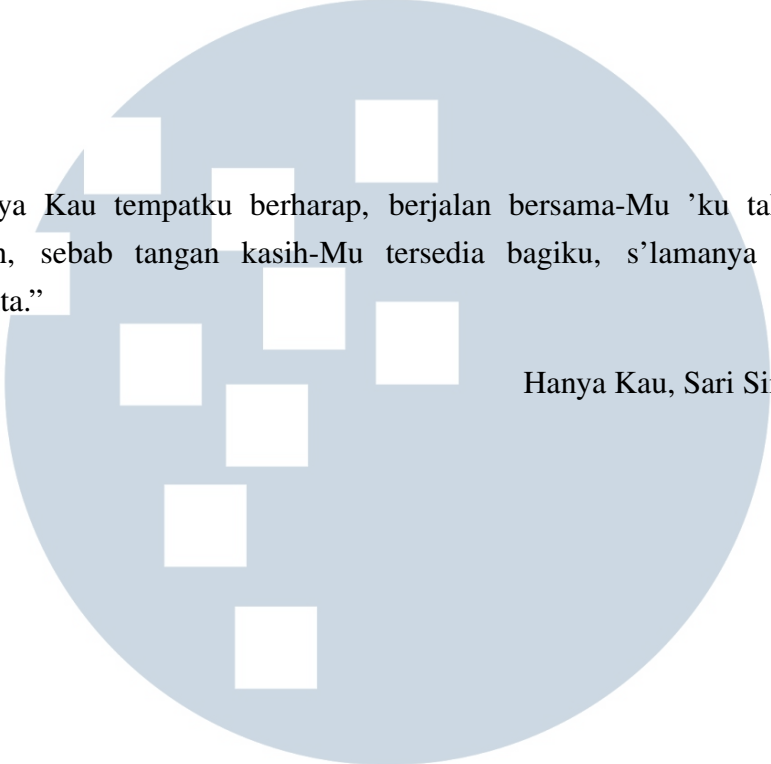
Tangerang, 13 November 2025

Yang menyatakan



Caroline Susanto

HALAMAN PERSEMBAHAN / MOTTO



"Hanya Kau tempatku berharap, berjalan bersama-Mu 'ku takkan goyah, sebab tangan kasih-Mu tersedia bagiku, s'lamanya Kau kucinta."

Hanya Kau, Sari Simorangkir

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir berjudul “Pengembangan Alat Bantu Berbasis AI Agent untuk Deteksi Kerentanan Broken Access Control melalui Otomatisasi Pengujian Endpoint pada Aplikasi Back-End” dengan baik.

Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Informatika, Universitas Multimedia Nusantara (UMN). Penelitian berfokus pada perancangan dan implementasi AI Agent untuk otomatisasi pengujian *endpoint back-end* pada aplikasi *Employee Self Service* (ESS), khususnya dalam mendeteksi kerentanan *Broken Access Control*. Ucapan terima kasih penulis sampaikan kepada:

1. Bapak Dr. Ir. Andrey Andoko, M.Sc., selaku Rektor Universitas Multimedia Nusantara.
2. Bapak Dr. Eng. Niki Prastomo, S.T., M.Sc., selaku Dekan Fakultas Teknik dan Informatika Universitas Multimedia Nusantara.
3. Bapak Arya Wicaksana, S.Kom., M.Eng.Sc., OCA, CEH, CEI, selaku Ketua Program Studi Informatika Universitas Multimedia Nusantara.
4. Bapak Moeljono Widjaja, B.Sc., M.Sc., Ph.D, sebagai Pembimbing yang telah memberikan bimbingan, arahan, dan motivasi atas terselesainya tugas akhir ini.
5. Kepada Pembimbing di Perusahaan yang telah memberikan saran selama pelaksanaan tugas akhir.
6. Kepada Mentor pertama di Perusahaan yang telah memberikan bimbingan, saran, serta ilmu di bidang kecerdasan buatan dan keamanan.
7. Kepada Mentor kedua di Perusahaan yang telah memberikan bimbingan, saran, serta ilmu di bidang pengembangan *back-end*.
8. Kepada Ayah, Ibu, dan Adik penulis yang telah memberikan dukungan moral dan material sehingga penulis dapat menyelesaikan tugas akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki keterbatasan. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan demi perbaikan

di masa mendatang. Besar harapan penulis agar karya ini memberikan manfaat praktis bagi organisasi dalam meningkatkan keamanan aplikasi *back-end*, sekaligus kontribusi teoretis bagi pengembangan keilmuan di bidang keamanan aplikasi dan pengujian keamanan perangkat lunak dengan bantuan *Artificial Intelligence*.

Tangerang, 13 November 2025



Caroline Susanto



**PENGEMBANGAN ALAT BANTU BERBASIS AI AGENT UNTUK
DETEKSI KERENTANAN BROKEN ACCESS CONTROL
MELALUI OTOMATISASI PENGUJIAN ENDPOINT
PADA APLIKASI BACK-END**

Caroline Susanto

ABSTRAK

Proses pembaruan data pribadi karyawan di PT. XYZ dilakukan secara manual melalui *Human Resource* dan dinilai tidak efektif karena memiliki potensi keterlambatan, kesalahan, dan risiko kebocoran data. Untuk mengatasi hal tersebut, dibangun aplikasi internal *Employee Self Service* berbasis arsitektur *decoupled* dengan implementasi *Role-Based Access Control*. Namun, pengujian keamanan *endpoint back-end* masih dilakukan secara manual menggunakan *Postman* dengan keterbatasan *coverage* hanya 50% dari 56 *endpoint*, waktu bervariasi signifikan, dan dokumentasi tidak terstruktur. Aplikasi ESS memiliki 28 *endpoint* kritis yang rentan terhadap *Broken Access Control* seperti *Insecure Direct Object Reference* dan *Broken Object Level Authorization*. Penelitian ini mengembangkan BYEBAC, alat bantu berbasis AI Agent untuk otomatisasi pengujian *endpoint back-end* guna mendeteksi kerentanan BAC *horizontal* dan vertikal. BYEBAC mengintegrasikan *Large Language Model Gemini 3.0 Pro Preview* sebagai *reasoning engine* untuk menganalisis matriks RBAC, *OpenAPI Specification*, dan *policy rules*. Sistem terdiri dari enam komponen modular yang bekerja dalam siklus *plan-discover-execute-classify-report* untuk melakukan *automated test planning*, *resource ID discovery*, *multi-role testing*, dan *AI-powered security summary*. Hasil pengujian pada 28 *endpoint* kritis menunjukkan BYEBAC mencapai akurasi 97,3% dengan *precision* 91,4%, *recall* 100%, dan *F1-score* 95,5% dengan konsistensi 100% pada lima eksekusi independen. BYEBAC menguji 130 *test case* dalam waktu rata-rata dua menit dengan *coverage* 100%, 89 kali lebih cepat dibandingkan pengujian manual tiga jam yang hanya mencakup 50% *endpoint*. Sistem berhasil mendeteksi seluruh kerentanan BAC tanpa *false negative* dan menghasilkan dokumentasi terstruktur format JSON dan *Markdown* yang dapat diintegrasikan pada *CI/CD*, meningkatkan efektivitas pengujian keamanan aplikasi ESS secara signifikan.

Kata kunci: Agen AI, *Broken Access Control*, *OpenAPI Specification*, Pengujian Keamanan Otomatis, *RESTful API*

DEVELOPMENT OF AN AI AGENT-BASED TOOL FOR BROKEN ACCESS CONTROL VULNERABILITY DETECTION THROUGH ENDPOINT TESTING AUTOMATION IN BACK-END APPLICATIONS

Caroline Susanto

ABSTRACT

The employee personal data update process at PT. XYZ is conducted manually through Human Resources and is considered ineffective due to potential delays, errors, and risks of data breaches. To address this, an internal Employee Self Service application was built using a decoupled architecture with Role-Based Access Control implementation. However, back-end endpoint security testing is still performed manually using Postman with limited coverage of only 50% of 56 endpoints, significantly varying time, and unstructured documentation. The ESS application has 28 critical endpoints vulnerable to Broken Access Control, such as Insecure Direct Object Reference and Broken Object Level Authorization. This research develops BYEBAC, an AI Agent-based tool for automating back-end endpoint testing to detect horizontal and vertical BAC vulnerabilities. BYEBAC integrates Large Language Model Gemini 3.0 Pro Preview as a reasoning engine to analyze RBAC matrix, OpenAPI Specification, and policy rules. The system consists of six modular components working in a plan-discover-execute-classify-report cycle to perform automated test planning, resource ID discovery, multi-role testing, and AI-powered security summary. Testing results on 28 critical endpoints show BYEBAC achieves 97.3% accuracy with a precision 91.4%, recall 100%, and an F1-score 95.5% with 100% consistency across five independent executions. BYEBAC tests 130 test cases in average two minutes with 100% coverage, 89 times faster than manual testing, three hours covering only 50% endpoints. The system successfully detects all BAC vulnerabilities without false negatives and generates structured documentation in JSON and Markdown format ready for CI/CD integration, significantly improving ESS application security testing effectiveness.

Keywords: AI Agent, Automated Security Testing, Broken Access Control, OpenAPI Specification, RESTful API

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN TIDAK MELAKUKAN PLAGIAT	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	iv
HALAMAN PERSEMBAHAN/MOTO	v
KATA PENGANTAR	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR KODE	xiv
DAFTAR RUMUS	xv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Permasalahan	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	6
BAB 2 LANDASAN TEORI	7
2.1 Back-End	7
2.2 Large Language Model	7
2.3 LLM-Based Intelligent Test Planning	8
2.4 AI Agent	8
2.5 Orchestrator dalam Sistem AI Agent	8
2.6 OpenAPI Specification	9
2.7 REST API	9
2.8 Auto-Discovery Resource IDs	10
2.9 Rule-Based Pattern Algorithm	11
2.10 Policy-Based Algorithm	11
2.11 Broken Access Control	11
2.12 Role-Based Access Control	14
2.13 Confusion Matrix	15
2.14 Coverage	17
2.15 Evaluasi Model	17
BAB 3 METODOLOGI PENELITIAN	19
3.1 Metode Penelitian	19
3.1.1 Studi Literatur	20
3.2 Pengumpulan Data dan Spesifikasi Sistem	20
3.2.1 Pengumpulan Data	20
3.2.2 Spesifikasi Sistem	21
3.3 Analisis Kebutuhan Sistem	21
3.4 Perancangan Arsitektur dan Algoritma Sistem	23
3.4.1 Perancangan Arsitektur Sistem	24
3.4.2 Perancangan Algoritma Sistem	25
3.5 Implementasi Agen dan Integrasi Sistem	26

3.5.1	Implementasi Agen	26
3.5.2	Integrasi Sistem	28
3.6	Evaluasi Eksperimental dan Analisis Hasil	29
3.6.1	Evaluasi Eksperimental	29
3.6.2	Analisis Hasil	30
BAB 4	HASIL DAN DISKUSI	31
4.1	Ringkasan Implementasi	31
4.1.1	Perangkat Keras	31
4.1.2	Perangkat Lunak	31
4.2	Hasil Pengujian Otomatis	32
4.2.1	Konfigurasi Pengujian Otomatis	33
4.2.2	Hasil Eksekusi Lima Kali Pengujian Otomatis	33
4.2.3	Confusion Matrix Pengujian Otomatis	34
4.2.4	Performance Metrics Pengujian Otomatis	35
4.2.5	Coverage	37
4.2.6	Hasil Pengujian Manual	37
4.2.7	Konfigurasi Pengujian Manual	37
4.2.8	Hasil Eksekusi Pengujian Manual	38
4.2.9	Confusion Matrix Pengujian Manual	39
4.2.10	Performance Metrics Pengujian Manual	40
4.3	Perbandingan Pengujian Manual dan Otomatis	40
4.3.1	Perbandingan Waktu Eksekusi	41
4.3.2	Perbandingan Coverage	42
4.3.3	Perbandingan Konsistensi	42
4.3.4	Perbandingan Akurasi Deteksi	43
4.3.5	Perbandingan Response Code	45
4.3.6	Perbandingan Fitur	46
4.4	Analisis Peningkatan Efektivitas Pengujian Keamanan	46
4.4.1	Baseline Keamanan Sebelum Pengujian Otomatis	47
4.4.2	Peningkatan Keamanan Sesudah Pengujian Otomatis	47
4.4.3	Metrik Peningkatan	48
4.5	Diskusi	49
4.5.1	Analisis Hasil Pengujian	49
4.5.2	Perbandingan Pengujian Manual dan Otomatis	50
4.5.3	Keterbatasan Penelitian	50
BAB 5	KESIMPULAN DAN SARAN	51
5.1	Simpulan	51
5.2	Saran	53
	DAFTAR PUSTAKA	55

UNIVERSITAS
MULTIMEDIA
NUSANTARA

DAFTAR TABEL

Tabel 3.1	Klasifikasi hasil pengujian keamanan otomatis	30
Tabel 4.1	Hasil lima kali pengujian otomatis	33
Tabel 4.2	Confusion matrix pengujian otomatis	34
Tabel 4.3	Performance metrics pengujian otomatis	35
Tabel 4.4	10 daftar hasil uji coba manual	38
Tabel 4.5	Confusion matrix pengujian manual	39
Tabel 4.6	Performance metrics pengujian manual	40
Tabel 4.7	Perbandingan waktu eksekusi pengujian manual dan otomatis	41
Tabel 4.8	Perbandingan coverage pengujian	42
Tabel 4.9	Perbandingan konsistensi hasil	43
Tabel 4.10	Perbandingan akurasi deteksi	44
Tabel 4.11	Perbandingan 6 daftar respons endpoint manual dan otomatis	45
Tabel 4.12	Perbandingan fitur pengujian	46
Tabel 4.13	Metrik peningkatan efektivitas	48
Tabel A1	Hasil pengujian manual pada 28 endpoint	70
Tabel A2	Perbandingan 28 daftar respons endpoint manual dan otomatis	73
Tabel A3	Matriks RBAC berdasarkan role dan permission	73



DAFTAR GAMBAR

Gambar 2.1	Flowchart representational state transfer application programming interface	10
Gambar 2.2	Flowchart broken access control	12
Gambar 2.3	Flowchart horizontal broken access control	13
Gambar 2.4	Flowchart vertical broken access control	14
Gambar 2.5	Flowchart role-based access control	15
Gambar 3.1	Flowchart penelitian	19
Gambar 3.2	Alur proses pengumpulan dan implementasi BYEBAC . .	26



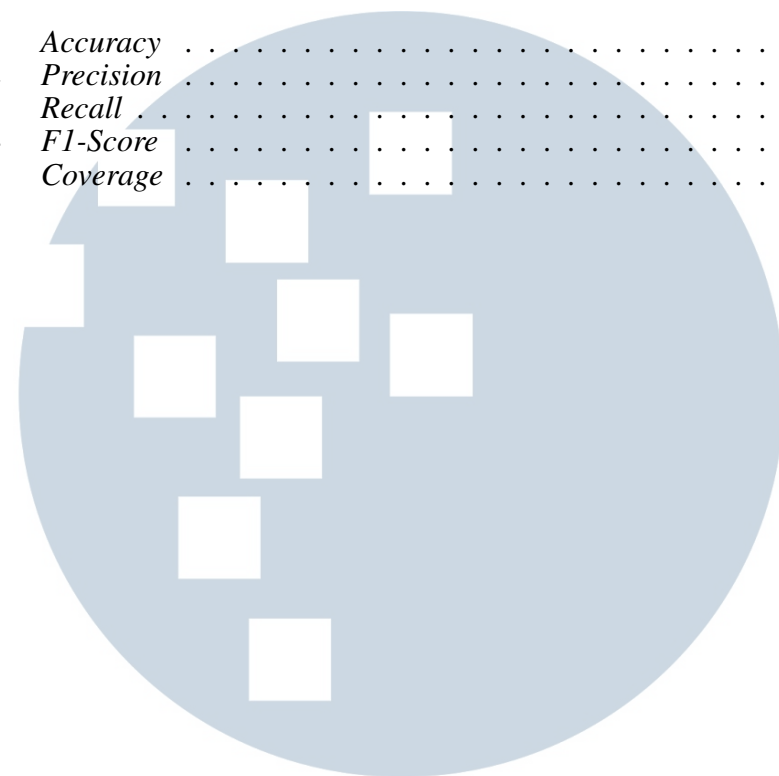
DAFTAR KODE

Kode A1	Kode python algoritma discovery IDs	69
Kode A2	System prompt - security test planner	70
Kode A3	System prompt - AI security testing agent	70
Kode A4	System prompt - BAC API test generator	71
Kode A5	System prompt - security triager	72
Kode A6	Contoh konfigurasi environment pengujian otomatis	72
Kode A7	Contoh OpenAPI Specification endpoint /auth/login	74
Kode A8	Policy rules endpoint admin dan employee	75
Kode A9	Laporan uji coba otomatis dalam markdown dengan AI summary	76
Kode A10	Laporan uji coba otomatis dalam JSON	77



DAFTAR RUMUS

Rumus 2.1	<i>Accuracy</i>	16
Rumus 2.2	<i>Precision</i>	16
Rumus 2.3	<i>Recall</i>	16
Rumus 2.4	<i>F1-Score</i>	16
Rumus 2.5	<i>Coverage</i>	17



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

DAFTAR LAMPIRAN

Lampiran 1	Hasil Persentase Turnitin	61
Lampiran 2	Formulir Bimbingan	67
Lampiran 3	Kode Algoritma Pengujian Otomatis	69
Lampiran 4	Hasil Pengujian Manual	70
Lampiran 5	Prompt Pengujian Otomatis AI Agent	70
Lampiran 6	Contoh Berkas Konfigurasi .env	72
Lampiran 7	Lampiran Perbandingan Hasil pengujian manual dan otomatis .	73
Lampiran 8	Data BYEBAC	73
Lampiran 9	Laporan Hasil Uji Coba Otomatis	76

