

PENGEMBANGAN EKOSISTEM
***STATIC APPLICATION SECURITY TESTING* UNTUK**
MENGIDENTIFIKASI KERENTANAN KEAMANAN SUMBER KODE
APLIKASI MENGGUNAKAN PENDEKATAN DEVSECOPS

TUGAS AKHIR

Diajukan sebagai salah satu syarat memperoleh gelar Sarjana
Program Studi S-1 Teknologi Informasi



Disusun oleh:

FAJAR SYAHRUDDIN

212104005

PROGRAM STUDI S-1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK & TEKNOLOGI INFORMASI
UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA
2025

HALAMAN PENGESAHAN

TUGAS AKHIR
PENGEMBANGAN EKOSISTEM
STATIC APPLICATION SECURITY TESTING UNTUK
MENGIDENTIFIKASI KERENTANAN KEAMANAN SUMBER KODE
APLIKASI MENGGUNAKAN PENDEKATAN DEVSECOPS

Diajukan oleh:

FAJAR SYAHRUDDIN

212104005

Telah dipertahankan di depan dewan penguji dan dinyatakan sah
sebagai salah satu syarat untuk memperoleh gelar Sarjana
di Fakultas Teknik & Teknologi Informasi
Universitas Jenderal Achmad Yani Yogyakarta

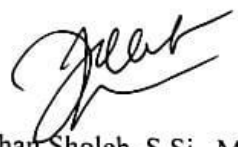
Tanggal:

Mengesahkan:

Pembimbing


Ir. Dedy Hariyadi, S.T., M.Kom
NIDN: 0518108001

Penguji


Adkhan Sholeh, S.Si., M.Cs
NIDN: 0510127501

Ketua Program Studi S-1 Teknologi Informasi
Fakultas Teknik & Teknologi Informasi
Universitas Jenderal Achmad Yani Yogyakarta



Rama Sahtyawan, S.T., M.Cs.
NPP:2019.13.0150

PERNYATAAN

Saya yang bertanda tangan di bawah ini, adalah mahasiswa Fakultas Teknik dan Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta,

Nama : Fajar Syahrudin
NPM : 212104005
Program Studi : S-1 Teknologi Informasi
Judul Tugas Akhir : Pengembangan Ekosistem *Static Application Security Testing* Untuk Mengidentifikasi Kerentanan Keamanan Sumber Kode Aplikasi Menggunakan Pendekatan Devsecops

Menyatakan bahwa hasil penelitian dengan judul tersebut di atas adalah asli karya saya sendiri dan bukan hasil plagiarisme. Semua referensi dan sumber terkait yang dikutip dalam karya ilmiah ini telah ditulis sesuai kaidah penulisan ilmiah yang berlaku. Dengan ini, saya menyatakan untuk menyerahkan hak cipta penelitian kepada Universitas Jenderal Achmad Yani Yogyakarta guna kepentingan ilmiah.

Demikian surat pernyataan ini dibuat dengan sebenar-benarnya tanpa ada paksaan dari pihak mana pun. Apabila terdapat kekeliruan atau ditemukan adanya pelanggaran akademik di kemudian hari, maka saya bersedia menerima konsekuensi yang berlaku sesuai ketentuan akademik.

Yogyakarta, 13 Juni 2025


Fajar Syahrudin

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul: “Pengembangan Ekosistem *Static Application Security Testing* Untuk Mengidentifikasi Kerentanan Keamanan Sumber Kode Aplikasi Menggunakan Pendekatan Devsecops”. Penyusunan laporan ini merupakan salah satu persyaratan untuk menyelesaikan studi di Program Studi S-1 Teknologi Informasi Fakultas Teknik & Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta. Laporan ini dapat diselesaikan atas bimbingan, arahan, dan bantuan dari berbagai pihak. Pada kesempatan ini penulis dengan rendah hati mengucapkan terima kasih dengan setulus-tulusnya kepada:

1. Bapak Rama Sahtyawan, S.T., M.Cs. selaku Dekan Fakultas Teknik dan Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta;
2. Bapak Ir. Dedy Hariyadi, S.T., M.Kom. selaku Dosen Pembimbing Tugas Akhir;
3. Bapak Chanief Budi Setiawan, S.T., M.Eng. selaku Dosen Pembimbing Akademik;
4. Para dosen yang telah memberikan banyak bekal ilmu pengetahuan kepada penulis selama menjadi mahasiswa di Fakultas Teknik dan Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta;
5. Ayah penulis, sebagai motivasi dan nasihat beliau yang selalu menjadi jalan hidup penulis, serta tak luput juga ibu penulis sebagai penyemangat, dan adik penulis yang menjadi tujuan menyelesaikan masa studi;
6. Rekan-rekan mahasiswa Prodi S-1 Teknologi Informasi angkatan 2021 di Universitas Jenderal Achmad Yani Yogyakarta yang sudah memberi dukungan dan kerja sama selama pembuatan tugas akhir.

Penulis menyadari bahwa laporan tugas akhir ini masih jauh dari kata sempurna. Maka dari itu dengan segala kerendahan hati penulis sangat menghargai

adanya kritik dan saran yang membangun dari semua pihak yang bersedia meluangkan waktu untuk membaca laporan tugas akhir ini.

Yogyakarta, 13 Juni 2025



Fajar Syahrudin

PERPUSTAKAAN
UNIVERSITAS JENDRAL ACHMAD YAN
YOGYAKARTA

DAFTAR ISI

Halaman judul	i
Halaman Pengesahan	ii
Pernyataan.....	iii
Kata Pengantar	iv
Daftar Isi.....	vi
Daftar Tabel	ix
Daftar Gambar.....	x
Daftar Lampiran.....	xii
Daftar Singkatan.....	xiii
Intisari.....	xiv
Abstract	xv
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Pertanyaan Penelitian	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Hasil Penelitian	5
Bab 2 Tinjauan Pustaka dan Landasan Teori	6
2.1 Tinjauan Pustaka	6
2.2 Landasan Teori.....	11
2.2.1 Software Development Life Cycle (SDLC).....	11
2.2.2 Secure Software Developymnt Life Cycle (SSDLC).....	13
2.2.3 Static Application Security Testing (SAST).....	17
2.2.4 DevSecOps.....	20
2.2.5 Continuous Integration/Continuous Deployment (CI/CD) ...	22
2.2.6 Container	29
Bab 3 Metode Penelitian.....	32
3.1 Secure Software Development Life Cycle (SSDLC).....	32
3.1.1 <i>Planning</i> (Perencanaan)	33

3.1.2	<i>Design</i> (Perancangan)	33
3.1.3	<i>Development</i> (Pengembangan).....	33
3.1.4	<i>Testing</i> (Pengujian)	34
3.1.5	<i>Release</i> (Rilis dan Pemeliharaan).....	34
3.2	Bahan Penelitian.....	34
3.3	Alat Penelitian.....	36
3.4	Perangkat Keras	36
3.5	Perangkat Lunak.....	37
3.6	Jalan Penelitian.....	37
3.6.1	Tahap Identifikasi dan Analisis.....	38
3.6.2	Tahap pengumpulan dan Pengolahan data	38
3.6.3	Tahap Perancangan dan Implementasi.....	39
3.6.4	Tahap Pengujian dan Evaluasi	39
3.6.5	Tahap Deployment dan Monitoring	40
Bab 4	Hasil Penelitian.....	41
4.1	Ringkasan Hasil Penelitian	41
4.2	Tahap Identifikasi dan Analisis.....	43
4.3	Tahap Pengumpulan dan Pengolahan Data.....	44
4.4	Tahap Perancangan dan Implementasi.....	45
4.4.1	Vulnerability Assessment (VA)	46
4.4.2	Static Application Security Testing (SAST).....	47
4.5	Tahap Pengujian dan Evaluasi	48
4.5.1	Vulnerability Assessment Menggunakan reNgin.....	49
4.5.2	Hasil dan Interpretasi dari VA Menggunakan reNgin.....	49
4.5.3	Pengujian SAST dengan SonarQube	50
4.5.4	Evaluasi Kerentanan.....	52
4.6	Tahapan Deployment dan Monitoring	70
4.6.1	Pemindaian ulang VA menggunakan reNgin.....	70
4.6.2	Pemindaian Ulang SAST menggunakan SonarQube.....	72
4.6.3	<i>Monitoring</i>	74
Bab 5	Kesimpulan dan Saran	77

5.1	Kesimpulan	77
5.2	Saran.....	78
	Daftar Pustaka	81
	Lampiran	84

PERPUSTAKAAN
UNIVERSITAS JENDRAL ACHMAD YANI
YOGYAKARTA

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka terdiri dari 5 sumber.....	10
Tabel 3.1 Sepesifikasi VPS Penelitian.....	37
Tabel 4.1 Penemuan VA menggunakan reNgin.....	50
Tabel 4.2 Hasil pengujian kerentanan pada SonarQube	51
Tabel 4.3 Hasil evaluasi kerentanan reNgin.....	53
Tabel 4.4 Tabel kerentanan setelah melakukan scan ulang	70
Tabel 4.5 Hasil kode kerentanan pada SonarQube setelah perbaikan	72

DAFTAR GAMBAR

Gambar 2.1 Metode SDLC (https://course-net.com/)	13
Gambar 2.2 Metode S-SDLC (https://www.digitalmaelstrom.net/)	17
Gambar 2.3 Fase DevSecOps (https://www.dynatrace.com/).....	21
Gambar 2.4 Metode CI/CD (https://www.blackduck.com/)	24
Gambar 2.5 Metode Kontainerisasi (https://www.docker.com/)	31
Gambar 3.1 Metode SSDLC (https://www.digitalmaelstrom.net/).....	33
Gambar 3.2 Jalan Penelitian.....	38
Gambar 4.1 Repository Penelitian	45
Gambar 4.2 Perancangan DevSecOps penelitian.....	46
Gambar 4.3 Tahapan <i>Vulnerability Assessment</i> dengan reNgin	47
Gambar 4.4 Diagram Arsitektur Sistem SAST	48
Gambar 4.5 Perbaikan kerentanan <i>Credentials Disclosure Check</i>	54
Gambar 4.6 Perbaikan kerentanan <i>Missing Subresource Integrity</i> (SRI) ..	55
Gambar 4.7 Perbaikan kerentanan <i>Hardcoded Credentials</i>	55
Gambar 4.8 Perbaikan kerentanan <i>hardcoded</i> pada Docker	56
Gambar 4.9 Perbaikan kerentanan <i>command injection</i>	57
Gambar 4.10 Perbaikan kerentanan CSRF pada <i>file root</i>	58
Gambar 4.11 Perbaikan kerentanan CSRF dengan <i>token Improve</i>	58
Gambar 4.12 Perbaikan kerentanan <i>code injection</i>	59
Gambar 4.13 Perbaikan kerentanan <i>insecure timeout</i>	59
Gambar 4.14 Perbaikan kerentanan <i>insecure user root</i>	60
Gambar 4.15 Perbaikan kerentanan <i>non root specify</i>	60
Gambar 4.16 Perbaikan konfigurasi <i>dcokerfile non root</i>	61
Gambar 4.17 Penambahan konfigurasi <i>.dockerignore</i>	62
Gambar 4.18 Perbaikan konfigurasi <i>dockerfile</i> dari ADD ke COPY	62
Gambar 4.19 Perbaikan konfigurasi <i>expose port</i> yang tidak dianjurkan ...	63
Gambar 4.20 Perbaikan alur koding <i>Weak Cryptography</i>	64
Gambar 4.21 Perbaikan alur koding <i>system random token</i> yang lemah	64
Gambar 4.22 Perbaikan <i>random secret token</i> karakter	65

Gambar 4.23 Perbaikan konfigurasi <i>expose debug</i>	66
Gambar 4.24 Perbaikan <i>hash alrogitm password</i> dengan bcrypt	67
Gambar 4.25 konfigurasi <i>unnecessary packages</i> pada dockerfile	68
Gambar 4.26 Perbaikan <i>Subresource Integrity</i> (SRI) kode HTML	68
Gambar 4.27 Perbaikan <i>algoritma md5 hashing</i>	69
Gambar 4.28 Perbaikan <i>technology server expose</i>	69
Gambar 4.29 Hasil <i>monitoring</i> setelah dilakukan <i>scan</i> ulang reNginx.....	71
Gambar 4.30 Dashboard Perbandingan hasil <i>Scanning</i>	71
Gambar 4.31 Terdapat Jumlah kode yang berstatus <i>Fixed</i>	74
Gambar 4.32 Hasil penurunan line grafik dilakukan perbaikan	75
Gambar 4.33 Hasil Tampilan projek SonarQube setelah diperbaiki.....	76

DAFTAR LAMPIRAN

Lampiran 1 Konfigurasi Dockerfile.....	84
Lampiran 2 Konfigurasi Jenkinsfile.....	86
Lampiran 3 Konfigurasi file nginx.conf	89
Lampiran 4 Konfigurasi file SonarQube proyek	91
Lampiran 5 Konfigurasi file .gitignore	92
Lampiran 6 Tutorial Instalasi dan Konfigurasi	94
Lampiran 7 Konfigurasi github webhook	106
Lampiran 8 Konfigurasi file server SonarQube.....	107
Lampiran 9 Konfigurasi Remote Docker Server di Jenkins	108
Lampiran 10 Konfigurasi Docker Server.....	109
Lampiran 11 Integrasi Github Melalui Jenkins	110
Lampiran 12 Perintah Remote Execute SSH dari Jenkins Ke Docker	111
Lampiran 13 Remote shell integrasi Jenkins ke Docker.....	112
Lampiran 14 Plugin SonarQube ang di install di Jenkins.....	113
Lampiran 15 Jadwal Penelitian.....	114
Lampiran 16 Lembar Bimbingan Dosen.....	115
Lampiran 17 Hasil Cek Plagiarisme	116

DAFTAR SINGKATAN

API_KEY	Application Programming Interface Key
CAMS	Culture, Automation, Measurement, and Sharing
CDN	Content Delivery Network
CFG	Control-flow graph
CI/CD	Continuous Integration/Continuous Deployment
CSRF	Cross-Site Request Forgery
CVSS	Common Vulnerability Scoring System
DAST	Dynamic Application Security Testing
DevSecOps	Development Security Operations
GDPR	General Data Protection Regulation
IAST	Interactive Application Security Testing
IDS	Intrusion Detection System
ISSAF	Information Systems Security Assessment Framework
OWASP	Open Web Application Security Project
RCE	Remote Code Execution
SAST	Static Application Security Testing
SATs	Static Analysis Tools
SCA	Software Composition Analysis
SDLC	Software Development Life Cycle
SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SRI	Subresource Integrity
SRS	Software Requirement Specification
SSDLC	Secure Software Development Life Cycle
TD	Technical Debt
TDD	Technical Debt Density
UML	Unified Modeling Language
VA	Vulnerability Assessment
VM	Virtual Machine
VPS	Virtual Private Server
XSS	Cross-site scripting