



**PERBANDINGAN EFISIENSI WAKTU DETEKSI (*MTTD*) DAN
WAKTU REMEDIASI (*MTTR*) ANTARA PENDEKATAN
KONVENSIONAL DAN *DEVSECOPS* PADA *PIPELINE*
GITOPS SISTEM PMB UIGM**

SKRIPSI

**Diajukan Sebagai Syarat untuk Menyelesaikan
Pendidikan Program Strata-1 Pada
Program Studi Teknik Informatika**

Billy Barasetia

2022110112

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER DAN SAINS
UNIVERSITAS INDO GLOBAL MANDIRI PALEMBANG**

2026

**PERBANDINGAN EFISIENSI WAKTU DETEKSI (*MTTD*) DAN
WAKTU REMEDIASI (*MTTR*) ANTARA PENDEKATAN
KONVENSIONAL DAN *DEVSECOPS* PADA *PIPELINE*
GITOPS SISTEM PMB UIGM**

SKRIPSI



Oleh:

Billy Barasetia

2022110112

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER DAN SAINS
UNIVERSITAS INDO GLOBAL MANDIRI PALEMBANG**

2026

LEMBAR PENGESAHAN SKRIPSI

LEMBAR PENGESAHAN SKRIPSI

Perbandingan Efisiensi Waktu Deteksi (MTTD) dan Waktu Remediasi (MTTR) antara Pendekatan Konvensional dan DevSecOps pada Pipeline GitOps Sistem PMB UIGM

Oleh

Billy Barasetia

NPM : 2022.110.112

Palembang , 2 September 2026

Pembimbing I



Dr. Herri Setiawan, M.Kom.
NIK : 2003.01.0060

Pembimbing II



Zaid Romegar Mair, S.T., M.Cs
NIK: 2021.01.0307

Mengetahui,

Dekan Fakultas Ilmu Komputer dan Sains



Dr. Herri Setiawan, M.Kom.
NIK: 2003.01.0060

LEMBAR PERSETUJUAN DEWAN PENGUJI

LEMBAR PERSETUJUAN DEWAN PENGUJI

Pada hari Jum'at tanggal 14 Agustus 2026 telah dilaksanakan ujian sidang skripsi :

Nama : Billy Barasetia

NPM : 2022.01.0112

Judul : Perbandingan Efisiensi Waktu Deteksi (*MTTD*) dan Waktu Remediasi (*MTTR*) antara Pendekatan Konvensional dan *DevSecOps* pada *Pipeline GitOps* Sistem PMB UIGM

Oleh Prodi Teknik Informatika Fakultas Ilmu Komputer dan Sains Universitas Indo Global Mandiri Palembang

Palembang, 19 Agustus 2026

Penguji 1,



Rudi Heriansyah, S.T., M.Eng., Ph.D

NIK: 2022.01.0315

Penguji 2,



Septa Cahyani, S.Kom., M.Cs

NIK: 1999.01.0006

Penguji 3,



Zaid Romegar Mair, S.T., M.Cs

NIK: 2021.01.0307

Menyetujui,
Ka. Prodi Teknik Informatika



Zaid Romegar Mair, S.T., M.Cs

NIK: 2021.01.0307

SURAT KETERANGAN REVISI SKRIPSI



SURAT KETERANGAN REVISI SKRIPSI
PROGRAM STUDI TEKNIK INFORMATIKA (S1)
FASILKOM DAN SAINS UNIVERSITAS INDO GLOBAL MANDIRI

Kami yang bertanda tangan dibawah ini, menerangkan bahwa :

Nama : Billy Barasetia
NPM : 2022.11.0112
Judul : Perbandingan Efisiensi Waktu Deteksi (*MTTD*) dan Waktu Remediasi (*MTTR*) antara Pendekatan Konvensional dan *DevSecOps* pada *Pipeline GitOps* Sistem PMB UIGM

Mahasiswa yang namanya tercantum diatas, telah selesai merevisi penulisan SKRIPSI

Palembang, 19 Agustus 2026

Penguji 1,

Rudi Heriansyah, S.T., M.Eng., Ph.D
NIK: 2022.01.0315

Penguji 2,

Septa Chayani, S.Kom., M.Cs
NIK: 2022.01.0358

Penguji 3,

Zaid Romegar Mair, S.T., M.Cs
NIK: 2021.01.0307

Menyetujui,
Ka. Prodi Teknik Informatika

Zaid Romegar Mair, S.T., M.Cs
NIK: 2021.01.0307

Perbandingan Efisiensi Waktu Deteksi (*MTTD*) dan Waktu Remediasi (*MTTR*) antara Pendekatan Konvensional dan *DevSecOps* pada *Pipeline GitOps* Sistem PMB UIGM

ABSTRAK

Penelitian ini membandingkan waktu deteksi dan waktu remediasi kerentanan antara pendekatan Konvensional dan *DevSecOps* pada *pipeline GitOps* sistem Penerimaan Mahasiswa Baru (PMB) Universitas Indo Global Mandiri (UIGM). Eksperimen terkontrol dilakukan melalui injeksi 10 sampel kerentanan yang diadaptasi dari dataset *DV-REST* dan diklasifikasikan berdasarkan *OWASP Top 10*. Metode *DevSecOps* menjalankan *SonarQube* sebagai *Static Application Security Testing (SAST)* dan *OWASP Dependency-Check* sebagai *Software Composition Analysis (SCA)* secara otomatis pada *pipeline Continuous Integration (CI)*, sedangkan metode Konvensional menjalankan pemindaian secara manual. Hasil perhitungan terhadap delapan kasus yang terdeteksi menunjukkan bahwa *DevSecOps* menghasilkan *MTTD* sebesar 11 menit 36,8 detik dan *MTTR* sebesar 22 menit 57,8 detik. Metode Konvensional menghasilkan *MTTD* sebesar 2 jam 52 menit 6,5 detik dan *MTTR* sebesar 24 jam 1 menit 48,3 detik. Dua kasus lainnya, yaitu A07 dan A10, tidak terdeteksi oleh instrumen yang digunakan. Berdasarkan 8 kasus yang terdeteksi, waktu deteksi pada pendekatan Konvensional 14,8 kali lebih lama dan waktu remediasinya 62,7 kali lebih lama dibandingkan pendekatan *DevSecOps*. Dengan demikian, pendekatan *DevSecOps* lebih efisien dalam kecepatan mendeteksi dan kecepatan remediasi pada kondisi eksperimen ini, tetapi pemindaian otomatis tetap perlu dilengkapi dengan pemeriksaan keamanan lainnya.

Kata Kunci: *DevSecOps*, *GitOps*, Keamanan Perangkat Lunak, Otomatisasi, *SCA*, *SAST*, Efisiensi Waktu.

Comparison of Mean Time to Detect (MTTD) and Mean Time to Remediate (MTTR) Between Conventional and DevSecOps Approaches in the GitOps Pipeline of the UIGM Student Admission System

ABSTRACT

This research compares vulnerability detection and remediation time between Conventional and DevSecOps approaches in the GitOps pipeline of the Universitas Indo Global Mandiri (UIGM) Student Admission System (PMB). A controlled experiment was conducted by injecting 10 vulnerability samples adapted from the DV-REST dataset and classified according to the OWASP Top 10. The DevSecOps method automatically ran SonarQube as Static Application Security Testing (SAST) and OWASP Dependency-Check as Software Composition Analysis (SCA) in the Continuous Integration (CI) pipeline, while the Conventional method used manual scanning. Based on eight detected cases, DevSecOps produced an MTTD of 11 minutes 36.8 seconds and an MTTR of 22 minutes 57.8 seconds. The Conventional method produced an MTTD of 2 hours 52 minutes 6.5 seconds and an MTTR of 24 hours 1 minute 48.3 seconds. Two other cases, A07 and A10, were not detected by the instruments used. Based on the 8 detected cases, the Conventional approach takes 14.8 times longer for detection and 62.7 times longer for remediation than the DevSecOps approach. Therefore, DevSecOps was more efficient for detection and remediation time under the experimental conditions, although automated scanning still needs to be complemented by other security checks.”

Keywords: *DevSecOps, GitOps, Software Security, Automation, SCA, SAST, Time Efficiency.*

KATA PENGANTAR

Puji dan Syukur Penulis persembahkan kepada Tuhan Yang Maha Esa sehingga penulis dapat menyelesaikan penyusunan laporan Seminar Proposal Skripsi yang berjudul “Perbandingan Efisiensi Waktu Deteksi (*MTTD*) dan Waktu Remediasi (*MTTR*) antara Pendekatan Konvensional dan *DevSecOps* pada *Pipeline GitOps* Sistem PMB UIGM”. Laporan ini disusun sebagai salah satu persyaratan akademik dalam proses penyusunan skripsi pada program studi yang penulis jalani. Sebagaimana tercantum pada Lampiran 1 mengenai Informasi Pribadi, penulis menyatakan bahwa seluruh data diri yang digunakan dalam laporan ini adalah benar dan dapat dipertanggungjawabkan.

Tidak lupa Penulis mengucapkan terima kasih atas bantuan yang diberikan selama penyusunan Kerja Praktek ini kepada:

1. Bapak Dr. Marzuki Alie, SE., MM., selaku Rektor Universitas Indo Global Mandiri Palembang.
2. Bapak Prof Dr Henderi, M.Kom sebagai Dekan Fakultas Ilmu Komputer.
3. Bapak Dr. Herri Setiawan, M.Kom sebagai Dosen Pembimbing I.
4. Bapak Zaid Romegar Mair, S.T., M.Cs., sebagai ketua Program Studi Teknik Informatika dan Dosen Pembimbing II.
5. Bapak Rendra Gustriansyah, S.T., M.Kom., sebagai Dosen Pembimbing Akademik.
6. Bapak/Ibu Dosen-dosen yang ada di Fakultas Ilmu Komputer Universitas Indo Global Mandiri.
7. Teman-teman di Lembaga Pengelolaan Teknologi Informasi dan Komunikasi (LPTIK)
8. Karyawan/karyawati di Fakultas Ilmu Komputer dan Sains Universitas Indo Global Mandiri
9. Orang tua yang memberikan doa dan dukungannya.
10. Teman-teman Teknik Informatika Angkatan 2022

Penulis menyadari bahwa penyusunan laporan Proposal Skripsi ini masih memiliki banyak kekurangan, oleh karena itu penulis mengharapkan saran dan kritik yang sifatnya membangun agar dapat digunakan demi perbaikan Proposal Skripsi ini nantinya. Penulis juga berharap agar Proposal Skripsi ini akan memberikan banyak manfaat bagi yang membacanya.

Palembang, 12 Agustus 2026

Penulis

Billy Barasetia

2022110112

DAFTAR ISI

HALAMAN JUDUL BAGIAN LUAR.....	I
HALAMAN JUDUL BAGIAN DALAM.....	II
LEMBAR PENGESAHAN SKRIPSI.....	III
LEMBAR PERSETUJUAN DEWAN PENGUJI.....	IV
SURAT KETERANGAN REVISI SKRIPSI.....	V
ABSTRAK.....	VI
ABSTRACT.....	VII
KATA PENGANTAR.....	VIII
DAFTAR ISI.....	X
DAFTAR GAMBAR.....	XIII
DAFTAR TABEL.....	XV
DAFTAR RUMUS.....	XVI
DAFTAR LAMPIRAN.....	XVII
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat.....	3
1.4.1 Tujuan.....	3
1.4.2 Manfaat.....	3
1.5 Metodologi Penelitian.....	4
1.5.1. Skenario dan Arsitektur.....	4
1.5.2. Teknik Perolehan Data.....	4
1.5.3. Parameter dan Metrik Pengukuran.....	5
1.5.4. Tahapan Pelaksanaan Penelitian.....	5
1.6 Sistematika Penulisan.....	6
BAB II KAJIAN PUSTAKA DAN DASAR TEORI.....	8
2.1 Tinjauan Pustaka.....	8
2.2 Evolusi DevOps ke DevSecOps dan Pendekatan Shift-Left.....	14
2.3 GitOps sebagai Kerangka Kerja Operasional.....	15
2.4 Standar Keamanan OWASP Top 10.....	15
2.5 Instrumen Pengujian Keamanan Otomatis.....	16
2.5.1 Software Composition Analysis (SCA).....	16
2.5.2 Static Application Security Testing (SAST).....	17
2.6 Ekosistem dan Instrumen Pendukung Eksperimen.....	18
2.6.1 GitHub Actions.....	18

2.6.2 SonarQube.....	19
2.6.3 OWASP Dependency-Check.....	20
2.6.4 National Vulnerability Database dan Common Vulnerabilities and Exposures.....	20
2.6.5 Dataset DV-REST (Damn Vulnerable REST API).....	20
2.7 Metrik Evaluasi Kinerja.....	21
2.7.1 Mean Time to Detect (MTTD):.....	21
2.7.2 Mean Time to Remediate (MTTR).....	21
2.8 Keterkaitan antara GitOps, DevSecOps, serta SCA dan SAST.....	22
2.9 Simbol Flowchart.....	25
BAB III METODE PENELITIAN.....	27
3.1 Kerangka Penelitian.....	27
3.2 Instrumen dan Dataset Pengujian.....	28
3.2.1 Infrastruktur CI/CD dan Repository Source Code.....	28
3.2.2 Instrumen Software Composition Analysis (SCA).....	29
3.2.3 Instrumen Static Application Security Testing (SAST).....	29
3.2.4 Dataset Kerentanan (DV-REST).....	30
3.3 Activity Diagram Eksperimen.....	34
3.4 Studi Literatur.....	36
3.5 Penyiapan Lingkungan Eksperimen (Konvensional & DevSecOps).....	36
3.5.1 Arsitektur Pendekatan Konvensional.....	36
3.5.2 Arsitektur Pendekatan DevSecOps.....	37
3.6 Pemilihan 10 Sampel Dataset DV-REST.....	38
3.7 Eksekusi Injeksi dan Pemindaian.....	39
3.8 Ekstraksi Data Waktu (Log Timestamp).....	40
3.9 Contoh Perhitungan MTTD dan MTTR.....	41
3.9.1 Perhitungan Mean Time to Detect (MTTD).....	41
3.9.2 Perhitungan Mean Time to Remediate (MTTR).....	42
3.10 Analisis Komparatif.....	43
BAB IV HASIL DAN PEMBAHASAN.....	45
4.1 Pengantar Hasil Eksperimen.....	45
4.1.1 Ruang Lingkup dan Metode.....	45
4.1.2 Konfigurasi Instrumen dan Workflow.....	45
4.1.3 Dasar Pengukuran dan Kelompok Analisis.....	47
4.2 Eksekusi Injeksi dan Pemindaian.....	49
4.2.1 A01: SSRF dan Penonaktifan Validasi TLS.....	51
4.2.2 A02: Password Database Kosong.....	52
4.2.3 A03: Library PyYAML Rentan.....	52

4.2.4 A04: Verifikasi Signature JWT Dinonaktifkan.....	54
4.2.5 A05: XML External Entity.....	55
4.2.6 A06: Konflik Body dan File pada FastAPI.....	56
4.2.7 A07: Password Reset Tanpa Rate Limiting.....	57
4.2.8 A08: XML Signature Tanpa Trusted Signer.....	57
4.2.9 A09: Pencatatan Data Sensitif.....	58
4.2.10 A10: Pengungkapan Traceback.....	61
4.3 Ekstraksi Data Waktu.....	61
4.3.1 Data TTD per Kasus.....	61
4.3.2 Data TTR DevSecOps.....	64
4.3.3 Data TTR Konvensional.....	65
4.4 Analisis Komparatif.....	67
4.4.1 Perbandingan MTTD Utama.....	67
4.4.2 Perbandingan MTTD Lima Kasus dengan Temuan Langsung.....	67
4.4.3 Kasus dengan Catatan dan Tidak Terdeteksi.....	68
4.4.4 Perbandingan MTTR Utama.....	68
4.4.5 Perbandingan MTTR Lima Kasus dengan Temuan Langsung.....	68
4.5 Pembahasan.....	69
4.5.1 Pengaruh Otomatisasi terhadap Deteksi.....	69
4.5.2 Cakupan Deteksi dan Kesesuaian Instrumen.....	69
4.5.3 Analisis Kasus A07 dan A10 yang Tidak Terdeteksi.....	69
4.5.4 Implementasi, Verifikasi, dan Quality Gate.....	72
4.5.5 Implikasi bagi Pengembangan Aplikasi PMB.....	73
4.5.6 Interpretasi Besaran Perbandingan.....	73
4.6 Validitas dan Keterbatasan Penelitian.....	73
4.6.1 Validitas Internal.....	73
4.6.2 Keterbatasan Scanner.....	74
4.6.3 Presisi Waktu dan Desain Dataset.....	74
4.6.4 Reproduksi dan Sumber Bukti.....	74
BAB V KESIMPULAN.....	78
5.1 Kesimpulan.....	78
5.2 Saran.....	78
DAFTAR PUSTAKA.....	80
DAFTAR LAMPIRAN.....	83

DAFTAR GAMBAR

Gambar 2.1 <i>Open Worldwide Application Security Project (OWASP)</i>	16
Gambar 2.2 Alur Kerja <i>Software Composition Analysis (SCA)</i>	17
Gambar 2.3 Alur Kerja <i>Static Application Security Testing (SAST)</i>	18
Gambar 2.4 <i>GitHub Actions</i>	19
Gambar 2.5 <i>SonarQube</i>	19
Gambar 2.6 <i>Framework Penelitian</i>	23
Gambar 2.7 <i>Flowchart Penelitian</i>	24
Gambar 3.1 Kerangka Tahapan Penelitian.....	27
Gambar 3.2 <i>Repository GitHub</i>	37
Gambar 3.3 <i>Repository Source Code DV-REST</i>	30
Gambar 3.4 Logo dan Deskripsi <i>Damn Vulnerable Restaurant</i>	31
Gambar 3.5 Potongan Kode <i>file get_order_status.py</i>	31
Gambar 3.6 <i>File get_order_status.py dalam dataset DV-REST</i>	32
Gambar 3.7 Potongan Kode PMB UIGM Yang asli.....	32
Gambar 3.8 Potongan Kode PMB UIGM Setelah di-injeksi kerentanan.....	33
Gambar 3.9 Tampilan <i>Web SonarQube</i> dan Membuka File admin.py	33
Gambar 3.10 <i>Activity Diagram</i>	35
Gambar 3.11 Diagram Arsitektur Eksperimen.....	38
Gambar 4.1 <i>File Workflow GitHub Actions DevSecOps</i>	46
Gambar 4.2 Daftar <i>Workflow GitHub Actions DevSecOps</i>	48
Gambar 4.3 Detail proses <i>SonarQube</i> pada <i>workflow CI</i>	49
Gambar 4.4 Detail proses <i>Dependency-Check</i> pada <i>workflow CI</i>	49
Gambar 4.5 Perubahan <i>source code</i> untuk remediasi A01.....	52
Gambar 4.6 Perubahan <i>source code</i> untuk remediasi A02.....	52
Gambar 4.7 Hasil laporan <i>HTML Dependency-Check</i> untuk A03.....	53
Gambar 4.8 Perubahan <i>source code</i> untuk remediasi A03.....	53
Gambar 4.9 Contoh temuan pada <i>SonarQube</i>	54
Gambar 4.10 Perubahan <i>source code</i> untuk remediasi A04.....	54
Gambar 4.11 Perubahan <i>source code</i> untuk remediasi A05.....	55
Gambar 4.12 Perubahan <i>source code</i> untuk remediasi A06.....	56

Gambar 4.13 Perubahan <i>source code</i> untuk remediasi A08.....	58
Gambar 4.14 Perubahan <i>source code</i> untuk remediasi A09.....	60
Gambar 4.15 Perbedaan Penyebab A07 dan A10 Tidak Terdeteksi.....	70

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu.....	8
Tabel 2.2 Simbol dan Fungsi Flowchart.....	25
Tabel 3.1 Skenario Injeksi Kerentanan A01-A10 <i>OWASP Top 10</i>	39
Tabel 4.1 Deskripsi, lokasi, dan hasil pemindaian A01-A10.....	50
Tabel 4.2 <i>TTD</i> Metode <i>DevSecOps</i>	61
Tabel 4.3 <i>TTD</i> Metode Konvensional.....	63
Tabel 4.4 <i>TTR</i> Metode <i>DevSecOps</i>	64
Tabel 4.5 <i>TTR</i> Metode Konvensional.....	66
Tabel 4.6 Perbandingan <i>MTTD</i> utama.....	67
Tabel 4.7 Perbandingan <i>MTTD</i> lima kasus dengan temuan langsung.....	67
Tabel 4.8 Perbandingan <i>MTTR</i> utama	68
Tabel 4.9 Perbandingan <i>MTTR</i> 5 kasus dengan temuan langsung	68
Tabel 4.10 Perbandingan A07 dan A10 yang Tidak Terdeteksi.....	72
Tabel 4.11 Sumber Bukti Eksperimen.....	75
Tabel 4.12 Sumber Bukti Konvensional dan Kriteria Perhitungan.....	76

DAFTAR RUMUS

2.1 <i>MTTD</i>	21
2.2 <i>MTTR</i>	22
3.1 Perhitungan <i>MTTD</i> Konvensional.....	41
3.2 Perhitungan <i>MTTD DevSecOps</i>	42
3.3 Peningkatan <i>MTTD</i> Antara <i>DevSecOps</i> dan Konvensional.....	42
3.4 Perhitungan <i>MTTR Konvensional</i>	42
3.5 Perhitungan <i>MTTR DevSecOps</i>	43
3.6 Peningkatan <i>MTTR</i> Antara <i>DevSecOps</i> dan Konvensional.....	43

DAFTAR LAMPIRAN

Lampiran 1. Daftar Riwayat Hidup.....	83
Lampiran 2. Surat Keterangan Siap Sidang Skripsi.....	84
Lampiran 3. Surat Persetujuan Ujian Skripsi.....	85
Lampiran 4. Surat Pernyataan Tidak Plagiat.....	86
Lampiran 5. Kartu Bimbingan Skripsi Bagian Depan.....	87
Lampiran 6. Kartu Bimbingan Skripsi Bagian Belakang.....	88