

**DETEKSI KERENTANAN KODE GENERATIVE AI
MENGUNAKAN GRAPHCODEBERT DENGAN
PENDEKATAN WEAK SUPERVISION**

Skripsi



**OLEH:
MUHAMMAD SHANDY DAILAMI
NIM. 2210817210013**

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
BANJARMASIN, JUNI 2026**

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Shandy Dailami
NIM : 2210817210013
Fakultas : Teknik
Prodi : Teknologi Informasi
Judul : Deteksi Kerentanan Kode *Generative AI* menggunakan GraphCodeBERT dengan Pendekatan *Weak Supervision*
Pembimbing Utama : Dr. Ir. Yuslena Sari, S.Kom., M.Kom., IPM.

Dengan ini saya menyatakan bahwa dalam Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar akademik di suatu perguruan tinggi, dan sepanjang pengetahuan saya, juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar rujukan.

Banjarmasin, 23 Juni 2025



M. SHANDY DAILAMI

NIM. 221081721001

LEMBAR PENGESAHAN

SKRIPSI PROGRAM STUDI S-1 TEKNOLOGI INFORMASI

Deteksi Kerentanan Kode *Generative AI* menggunakan GraphCodeBERT dengan
Pendekatan *Weak Supervision*

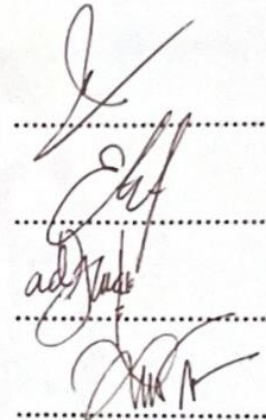
Oleh

Muhammad Shandy Dailami (2210817210013)

Telah dipertahankan di depan Tim Penguji pada 23 Juni 2026 dan dinyatakan

LULUS

Komite Penguji :
Ketua : Ir. Muhammad Alkaff, S.Kom., M.Kom., Ph.D
NIP. 198606132015041001
Anggota 1 : Erika Maulidiya, S.Kom., M.Kom.
NIP. 200006192025062016
Anggota 2 : Achmad Mujaddid Islami, S.Kom., M.Kom.
NIP. 199807102025061010
Pembimbing : Dr. Ir. Yuslena Sari, S.Kom., M.Kom., IPM.
Utama NIP. 198411202015042002



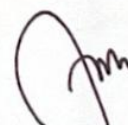
Banjarbaru, 10 JUL 2026

Diketahui dan disahkan oleh:

Wakil Dekan Bidang Akademik
Fakultas Teknik ULM,

Koordinator Program Studi
S-1 Teknologi Informasi,




Andrean Rizky Baskara, S.Kom., M.Kom.
NIP. 199307032019031011

PERSETUJUAN MAJU SIDANG SKRIPSI

**DETEKSI KERENTANAN KODE GENERATIVE AI MENGGUNAKAN
GRAPHCODEBERT DENGAN PENDEKATAN WEAK SUPERVISION**

OLEH

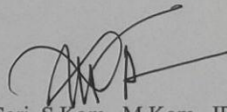
MUHAMMAD SHANDY DAILAMI

NIM. 2210817210013

Telah diperiksa dan terpenuhi semua persyaratan akademik, administrasi, dan disetujui untuk
dipertahankan di hadapan dewan penguji

Banjarmasin, 15 Juni 2026

Pembimbing Utama,



Dr. Ir. Yuslena Sari, S.Kom., M.Kom., IPM.

NIP. 198411202015042002

ABSTRAK

Pergeseran paradigma pengembangan perangkat lunak ke arah *vibe coding* yang memanfaatkan *Large Language Model* (LLM) menimbulkan risiko keamanan, karena kode yang dihasilkan AI terbukti rentan terhadap kelemahan keamanan. Metode deteksi konvensional seperti *Static Application Security Testing* (SAST) memiliki keterbatasan berupa tingkat *false positive* yang tinggi, sementara model *deep learning* yang ada cenderung gagal melakukan generalisasi lintas domain. Penelitian ini bertujuan membangun dataset *hybrid* serta melatih model GraphCodeBERT untuk mendeteksi kerentanan pada kode hasil *vibe coding*. Dataset dibangun dengan menggabungkan 50.052 sampel kode manusia berbahasa C/C++ dari Devign dan ReVeal yang telah divalidasi pakar, dengan 51.310 sampel kode Java hasil generasi AI yang dilabeli secara otomatis menggunakan teknik *Majority Vote* dari tiga alat SAST (CodeQL, SonarQube, dan VCG) dalam kerangka *Weakly Supervised Learning*. Model dilatih menggunakan strategi *continual learning* dua tahap dengan *micro learning rate* untuk mengatasi *catastrophic forgetting*. Hasil pengujian menunjukkan bahwa seluruh model baseline yang dilatih pada dataset tunggal gagal melakukan generalisasi lintas domain (F1-Score 0,000), sedangkan model *mixed final* mampu mendeteksi kerentanan pada tiga domain sekaligus dengan F1-Score 0,691, *precision* 0,715, dan *recall* 0,670 pada domain kode *vibe coding*, serta F1-Score 0,612 pada Devign. Mekanisme *majority vote* juga terbukti efektif meredam *false positive* SAST tunggal. Penelitian ini menyimpulkan bahwa strategi penggabungan dataset yang diusulkan menghasilkan model dengan kemampuan generalisasi yang lebih baik dibandingkan pelatihan pada dataset tunggal.

Kata kunci: *Catastrophic Forgetting*, Deteksi Kerentanan, *GraphCodeBERT*, *Vibe Coding*, *Weakly Supervised Learning*

ABSTRACT

The paradigm shift in software development toward vibe coding, which relies on Large Language Models (LLM), introduces security risks, as AI-generated code has been shown to be prone to security weaknesses. Conventional detection methods such as Static Application Security Testing (SAST) suffer from high false positive rates, while existing deep learning models tend to fail at cross-domain generalization. This study aims to build a hybrid dataset and train a GraphCodeBERT model to detect vulnerabilities in vibe-coding-generated code. The dataset was constructed by combining 50,052 expert-validated human C/C++ code samples from Devign and ReVeal with 51,310 AI-generated Java samples automatically labeled using an Majority Vote technique from three SAST tools (CodeQL, SonarQube, and VCG) within a Weakly Supervised Learning framework. The model was trained using a two-stage continual learning strategy with a micro learning rate to mitigate catastrophic forgetting. The results show that all baseline models trained on a single dataset failed to generalize across domains (F1-Score 0.000), whereas the mixed final model was able to detect vulnerabilities across three domains simultaneously, achieving an F1-Score of 0.691, precision of 0.715, and recall of 0.670 on the vibe coding domain, and an F1-Score of 0.612 on Devign. The majority vote mechanism also proved effective in suppressing the false positives of individual SAST tools. This study concludes that the proposed dataset-combining strategy yields a model with better generalization capability than training on a single dataset.

Keywords: Catastrophic Forgetting, Vulnerability Detection, GraphCodeBERT, Vibe Coding, Weakly Supervised Learning

KATA PENGANTAR

Puji dan syukur senantiasa kami panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, kesehatan, dan kekuatan, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Deteksi Kerentanan Kode *Generative AI* Menggunakan GraphCodeBERT dengan Pendekatan *Weak Supervision*” ini dengan baik. Shalawat serta salam semoga tercurah kepada junjungan kita, Nabi Besar Muhammad SAW, yang telah membawa kita dari zaman kegelapan menuju cahaya keilmuan.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata-1 (S1) pada Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Lambung Mangkurat, Banjarmasin. Penulis menyadari bahwa penyelesaian skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan terima kasih, penulis menyampaikan penghargaan kepada:

1. Bapak Dekan Fakultas Teknik Universitas Lambung Mangkurat, yang telah memberikan izin dan dukungan dalam proses penyusunan skripsi ini.
2. Bapak Ketua Jurusan Teknologi Informasi, yang selalu memberikan arahan dalam proses akademik.
3. Bapak Andreyan Rizky Baskara, S.Kom., M.T., selaku Koordinator Program Studi Teknologi Informasi, yang senantiasa memberikan bimbingan dan dukungan selama masa studi.
4. Ibu Dr. Ir. Yuslena Sari, S.Kom., M.Kom., IPM., selaku pembimbing utama, yang dengan penuh kesabaran telah memberikan arahan, bimbingan, dan motivasi kepada penulis.
5. Orang tua dan keluarga, yang selalu memberikan doa, dukungan moral, dan materiil tanpa henti.
6. Teman-teman seperjuangan di Program Studi Teknologi Informasi, yang telah menjadi tempat berbagi ilmu dan pengalaman selama masa perkuliahan.
7. Ikhtiara Fakhrunisa (2210817120003), sosok istimewa yang tak pernah lelah memberikan dukungan penuh, dan ruang aman bagi penulis untuk

bertukar cerita hingga skripsi ini dapat diselesaikan dengan baik.

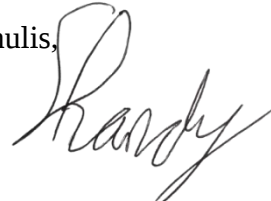
Skripsi ini terdiri dari beberapa bab dengan sistematika penulisan sebagai berikut:

- BAB I: Pendahuluan, yang mencakup latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian.
- BAB II: Tinjauan Pustaka, berisi landasan teori, penelitian terkait, dan kerangka pemikiran.
- BAB III: Metodologi Penelitian, meliputi alat dan bahan, lokasi penelitian, alur penelitian.
- BAB IV: Hasil dan Pembahasan, yang memaparkan hasil penelitian serta analisisnya.
- BAB V: Kesimpulan dan Saran, yang berisi kesimpulan dari hasil penelitian serta saran untuk pengembangan lebih lanjut.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun dari berbagai pihak sangat penulis harapkan untuk perbaikan di masa mendatang. Semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan menjadi referensi bagi para pembaca.

Banjarmasin, 23 Juni 2026

Penulis,



M. SHANDY DAILAMI

DAFTAR ISI

PERNYATAAN.....	i
LEMBAR PENGESAHAN	ii
PERSETUJUAN MAJUU SIDANG SKRIPSI.....	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN.....	v
DAFTAR RIWAYAT HIDUP.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah.....	6
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
BAB II TINJAUAN PUSTAKA.....	8
2.1 Penelitian Terkait.....	8
2.1.1 <i>GraphCodeBERT: Pre-Training Code Representations with Data Flow</i>	8
2.1.2 <i>Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions</i>	8
2.1.3 <i>On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in</i>	

<i>Web Applications</i>	9
2.1.4 <i>A Brief Introduction to Weakly Supervised Learning</i>	10
2.1.5 <i>Combining Datasets to Improve Model Fitting</i>	10
2.2 Landasan Teori	16
2.2.1 <i>Vibe Coding</i>	16
2.2.2 <i>Vulnerability</i>	16
2.2.3 CWE dan OWASP	17
2.2.4 <i>Static Analysis Security Tools (SAST)</i>	17
2.2.5 <i>Large Language Model</i>	18
2.2.6 <i>Weakly Supervised Learning</i>	18
2.2.7 <i>Majority Vote</i>	18
2.2.8 GraphCodeBERT	19
2.3 Kerangka Pemikiran	19
BAB III METODOLOGI PENELITIAN.....	22
3.1 Alat dan Bahan	22
3.1.1 Alat.....	22
3.1.2 Bahan	23
3.2 Lokasi dan Objek Penelitian.....	23
3.3 Alur Penelitian.....	24
3.3.1 Identifikasi Masalah	24
3.3.2 Studi Literatur	24
3.3.3 Akuisisi Data.....	25
3.3.4 Labeling Data	28
3.3.5 Pembuatan Model	34
3.3.6 Evaluasi Hasil	37

BAB IV HASIL DAN PEMBAHASAN	39
4.1 Alur Kerja Penelitian	39
4.2 Akuisisi dan Pengolahan Data	39
4.2.1 Dataset Kode Hasil Manusia	40
4.2.2 Dataset Kode Hasil AI	41
4.3 Pembuatan dan Pelatihan Model	44
4.3.1 Konfigurasi Model	44
4.3.2 Strategi Pelatihan	44
4.3.3 Hasil Pelatihan Model	45
4.4 Hasil dan Evaluasi Model	51
4.4.1 Hasil Pengujian Skenario Baseline	51
4.4.2 Hasil Pengujian Skenario Mixed Dataset	60
4.4.3 Analisis Kesalahan Prediksi	67
4.4.4 Komparasi Model pada Dataset Sintetis Java	71
4.5 Pembahasan	74
4.5.1 Efektivitas Strategi Mixed Dataset	74
4.5.2 Analisis Kinerja pada Reveal	75
4.5.3 Perbandingan dengan Pendekatan SAST	76
BAB V KESIMPULAN DAN SARAN	77
5.1 Kesimpulan	77
5.2 Saran	78
DAFTAR PUSTAKA	79
LAMPIRAN	84

DAFTAR GAMBAR

Gambar 2.1 Arsitektur GraphCodeBERT	19
Gambar 2.2 Kerangka Pemikiran	20
Gambar 3.1 Alur Penelitian	24
Gambar 3.2 Grafik Jumlah Data Kode.....	25
Gambar 3.3 Proses Analisis CodeQL	29
Gambar 3.4 Output Hasil Analisis CodeQL	30
Gambar 3.5 Proses Analisis SonarQube.....	31
Gambar 3.6 Output Hasil Analisis SonarQube	32
Gambar 3.7 Output Hasil Analisis VCG	32
Gambar 3.8 Alur Pembuatan Model	36
Gambar 4.1 Metode Pengumpulan Data	41
Gambar 4.3 Grafik Pelatihan Model GraphCodeBERT Menggunakan Dataset Devign .	46
Gambar 4.4 Grafik Pelatihan Model GraphCodeBERT Menggunakan Dataset ReVeal .	47
Gambar 4.5 Grafik Pelatihan Model GraphCodeBERT Menggunakan Dataset Sintetis Java	48
Gambar 4.6 Grafik Pelatihan Model GraphCodeBERT Menggunakan Dataset <i>Mixed</i> Devign dan ReVeal.....	49
Gambar 4.7 Grafik Pelatihan Model GraphCodeBERT Menggunakan Dataset <i>Mixed</i> Devign, ReVeal, dan Sintetis Java.....	50
Gambar 4.8 <i>Confusion Matrix Baseline</i> ReVeal diuji pada ReVeal <i>test set</i>	52
Gambar 4.9 <i>Confusion Matrix Baseline</i> ReVeal diuji pada Devign <i>test set</i>	53
Gambar 4.10 <i>Confusion Matrix Baseline</i> ReVeal diuji pada Sintetis Java <i>test set</i>	54
Gambar 4.11 <i>Confusion Matrix Baseline</i> Devign diuji pada Devign <i>test set</i>	55
Gambar 4.12 <i>Confusion Matrix Baseline</i> Devign diuji pada ReVeal <i>test set</i>	56
Gambar 4.13 <i>Confusion Matrix Baseline</i> diuji pada Sintetis Java	57

Gambar 4.14 <i>Confusion Matrix Baseline</i> diuji pada Sintetis Java	58
Gambar 4.15 <i>Confusion Matrix Baseline</i> Sintetis Java diuji pada Devign dan ReVeal ..	59
Gambar 4.16 <i>Confusion Matrix Stage 1</i> diuji pada Devign <i>test set</i>	61
Gambar 4.17 <i>Confusion Matrix Stage 1</i> diuji pada ReVeal <i>test set</i>	62
Gambar 4.18 <i>Confusion Matrix Mixed Stage 1</i> diuji pada Sintetis Java	63
Gambar 4.19 <i>Confusion Matrix Mixed Final</i> diuji pada Devign <i>test set</i>	64
Gambar 4.20 <i>Confusion Matrix Mixed Final</i> diuji pada ReVeal <i>test set</i>	65
Gambar 4.21 <i>Confusion Matrix Mixed Final</i> diuji pada Sintetis Java <i>test set</i>	66
Gambar 4.22 <i>Error Analysis</i> Distribusi Token <i>Threshold</i> > 10 pada Domain Devign...	68
Gambar 4.23 <i>Error Analysis</i> Distribusi Token pada Domain ReVeal.....	69
Gambar 4.24 <i>Error Analysis</i> Distribusi Token pada Domain Sintetis Java	70
Gambar 4.25 Grafik Perbandingan 8 Model.....	72

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait.....	12
Tabel 3.1 Daftar Alat Penelitian.....	22
Tabel 3.2 Daftar Bahan Penelitian.....	23
Tabel 3.3 Sampel Dataset Hasil Manusia.....	25
Tabel 3.4 <i>Prompt</i> dari Penelitian FormAI.....	26
Tabel 3.5 <i>Prompt Generate</i> Kode hasil AI.....	27
Tabel 3.6 Hasil <i>Prompt Generate</i> Kode Dari AI.....	27
Tabel 3.7 Sampel Kode <i>Generate AI</i> Pada <i>Majority Vote</i>	33
Tabel 3.8 Uji Skenario	35
Tabel 4.1 Distribusi Dataset Kode Hasil Manusia.....	40
Tabel 4.2 Contoh Kode Hasil AI.....	42
Tabel 4.3 Hasil Pelabelan SAST pada Dataset Kode Hasil AI	43
Tabel 4.4 Distribusi Hasil <i>Vote</i> Antar SAST pada Dataset AI	43
Tabel 4.5 <i>Hyperparameter</i> Pelatihan Skenario <i>Baseline</i>	44
Tabel 4.6 <i>Hyperparameter</i> Pelatihan Skenario <i>Mixed Dataset</i>	45
Tabel 4.7 Hasil Pengujian Lintas Domain untuk Skenario <i>Baseline</i>	51
Tabel 4.8 <i>Confusion Matrix Baseline</i> ReVeal ke ReVeal	52
Tabel 4.9 <i>Confusion Matrix Baseline</i> ReVeal ke Devign	53
Tabel 4.10 <i>Confusion Matrix Baseline</i> ReVeal ke Sintetis Java.....	54
Tabel 4.11 <i>Confusion Matrix Baseline</i> Devign ke Devign.....	55
Tabel 4.12 <i>Confusion Matrix Baseline</i> Devign ke ReVeal.....	56
Tabel 4.13 <i>Confusion Matrix Baseline</i> Devign ke Sintetis Java	57
Tabel 4.14 <i>Confusion Matrix Baseline</i> Sintetis Java ke Sintetis Java.....	58
Tabel 4.15 <i>Confusion Matrix Baseline</i> Sintetis Java ke Devign	59

Tabel 4.16 <i>Confusion Matrix Baseline</i> Sintetis Java ke ReVeal	59
Tabel 4.17 <i>Confusion Matrix Mixed Stage 1</i> ke Devign	60
Tabel 4.18 <i>Confusion Matrix Mixed Stage 1</i> ke ReVeal	61
Tabel 4.19 <i>Confusion Matrix Mixed Stage 1</i> ke Sintetis Java.....	62
Tabel 4.20 <i>Confusion Matrix Mixed Final</i> ke Devign	64
Tabel 4.21 <i>Confusion Matrix Mixed Final</i> ke ReVeal.....	65
Tabel 4.22 <i>Confusion Matrix Mixed Final</i> ke Sintetis Java	66
Tabel 4.23 Perbandingan TP <i>Baseline</i> dengan <i>Mixed</i>	67
Tabel 4.24 Ringkasan Pola Token dan <i>Error Analysis</i>	71
Tabel 4.25 Daftar Model yang Dibandingkan.....	72

DAFTAR LAMPIRAN

Lampiran 1. Lembar Konsultasi	84
Lampiran 2. Verifikasi Data Split.....	85