Journal of Information Technology and Computer Science (INTECOMS)
Volume 9 Nomor 2, Tahun 2026
e-ISSN : 2614-1574
p-ISSN : 2621-3249

CAN TRANSFORMER MODELS DETECT SCAM TOKENS? A STUDY ON ERC-20 SMART CONTRACTS

DAPATKAH MODEL TRANSFORMER MENDETEKSI TOKEN SCAM? SEBUAH STUDI PADA SMART CONTRACT ERC-20

Andhi Saputro[1], Makhsun[2], Ahmad Musyafa[3]
Teknik Informatika S-2, Program Pascasarjana, Universitas Pamulang[1,2,3]
hello.andhisaputro@gmail.com[1], dosen00345@unpam.ac.id[2], dosen00668@unpam.ac.id[2]

ABSTRACT

The rapid growth of the blockchain ecosystem has led to the emergence of thousands of tokens on the ERC-20 network. While this phenomenon fosters financial innovation, it also increases the risk of fraud through smart contracts that conceal malicious mechanisms such as backdoors, blacklist bots, and fee manipulation. This study proposes an end-to-end Transformer-based classification approach to detect ERC-20 scam tokens using Solidity source code as the sole input feature. Three models are evaluated: CodeBERT (microsoft/codebert-base), RoBERTa (roberta-base), and GraphCodeBERT (microsoft/graphcodebert-base). The dataset consists of 60,000 ERC-20 contracts obtained from the ASSERT-KTH/DISL repository, with 30,000 contracts semi-automatically labeled using rule-based static code analysis across seven scam categories: Honeypot, High Tax, Balance Manipulation, Blacklist, Hidden Owner, Rug Pull, and Unlimited Mint In binary classification, GraphCodeBERT achieved the best performance with an F1-score of 0.9295 and an AUC of 0.9808. In multilabel classification, RoBERTa outperformed the others in terms of F1-score (0.8681), while GraphCodeBERT achieved the highest AUC (0.9672). The Blacklist label posed a significant challenge, with an F1-score ranging from 0.61 to 0.64 due to extreme class imbalance. The results demonstrate that Solidity source code representations learned by Transformer models are sufficiently informative to automatically distinguish between scam and legitimate contracts.

Keywords: *ERC-20, Smart Contract, Scam Detection, CodeBERT, GraphCodeBERT, RoBERTa, Multi-Label Classification, Transformer, Solidity, Blockchain Security*

ABSTRAK

Pertumbuhan pesat ekosistem blockchain telah melahirkan ribuan token pada jaringan ERC-20. Fenomena ini mendorong inovasi finansial, namun sekaligus meningkatkan risiko penipuan melalui smart contract yang menyembunyikan mekanisme berbahaya seperti backdoor, blacklist bot, dan manipulasi fee. Penelitian ini mengusulkan pendekatan klasifikasi berbasis Transformer secara end-to-end untuk mendeteksi token scam ERC-20 menggunakan kode sumber Solidity sebagai satu-satunya fitur masukan. Tiga model dievaluasi: CodeBERT (microsoft/codebert-base), RoBERTa (roberta-base), dan GraphCodeBERT (microsoft/graphcodebert-base). Dataset terdiri dari 60.000 kontrak ERC-20 yang diambil dari repositori ASSERT-KTH/DISL, dengan 30.000 kontrak dilabeli secara semi-otomatis menggunakan analisis kode statis berbasis aturan (rule-based) untuk tujuh jenis scam: Honeypot, High Tax, Balance Manipulation, Blacklist, Hidden Owner, Rug Pull, dan Unlimited Mint. Pada klasifikasi biner, GraphCodeBERT mencapai performa terbaik dengan F1-Score 0,9295 dan AUC 0,9808. Pada klasifikasi multilabel, RoBERTa unggul pada F1-Score (0,8681) sementara GraphCodeBERT unggul pada AUC (0,9672). Label Blacklist menjadi tantangan tersendiri dengan F1-Score hanya 0,61–0,64 akibat ketidakseimbangan kelas yang ekstrem. Hasil penelitian membuktikan bahwa representasi kode sumber Solidity melalui model Transformer sudah cukup informatif untuk membedakan kontrak scam dari kontrak legitim secara otomatis.

Kata Kunci: ERC-20, Smart Contract, Scam Detection, CodeBERT, GraphCodeBERT, RoBERTa, Multi-Label Classification, Transformer, Solidity, Blockchain Security

PENDAHULUAN

Perkembangan teknologi blockchain telah menciptakan perubahan mendasar dalam sektor keuangan digital, khususnya melalui kemunculan Decentralized Finance (DeFi) yang memungkinkan transaksi antar individu tanpa perantara institusional [1]. Salah satu tulang punggung sistem ini adalah smart contract pada jaringan Ethereum, yang menggerakkan ribuan token berbasis standar ERC-20. Standar ERC-20 memberikan kerangka kerja yang

konsisten sehingga token-token tersebut dapat saling berinteraksi dalam ekosistem Ethereum secara mudah, mendorong terciptanya jutaan token baru untuk keperluan ICO, DEX, NFT, hingga play-to-earn games.

Kemudahan menciptakan token baru membawa dampak negatif yang signifikan: meningkatnya jumlah scam token yang menyembunyikan kode berbahaya dalam kontrak yang tampak sah. Zhang [2] mencatat bahwa 260.000 token, atau 98% dari sekitar 2,5 juta smart contract di jaringan Ethereum, adalah token ERC-20, yang sekaligus menyoroti besarnya permukaan serangan standar ini. Xia et al. [3] mengungkapkan bahwa lebih dari 50% token yang diluncurkan di platform DEX seperti Uniswap memiliki karakteristik scam yang dimasukkan secara eksplisit ke dalam kode smart contract.

Metode deteksi konvensional, seperti audit manual dan sistem berbasis aturan (rule-based), memiliki keterbatasan kritis: lambat beradaptasi terhadap evolusi teknik scam, membutuhkan keahlian khusus, biaya tinggi, dan tidak skalabel terhadap volume kontrak yang terus bertambah [4]. Pendekatan berbasis machine learning, khususnya model Transformer seperti CodeBERT [5], GraphCodeBERT [6], dan RoBERTa [7], menawarkan alternatif yang lebih otomatis dan adaptif. Model-model ini telah pre-trained pada korpus kode sumber dan bahasa alami dalam jumlah besar, sehingga mampu menangkap representasi semantik kode secara kontekstual.

Penelitian ini berkontribusi dalam:

Menyajikan perbandingan sistematis tiga model Transformer (CodeBERT, RoBERTa, GraphCodeBERT) pada tugas deteksi scam token ERC-20 dalam dua skenario: klasifikasi biner dan klasifikasi multilabel tujuh kelas.; Membangun pipeline end-to-end dari kode sumber Solidity mentah hingga prediksi jenis scam tanpa rekayasa fitur manual.; serta Menganalisis tantangan khusus deteksi scam berbasis kode, termasuk ketidakseimbangan kelas dan variasi pola kode per kategori.

TINJAUAN PUSTAKA

Deteksi Scam pada Smart Contract

Chen et al. [8] melakukan analisis terhadap lebih dari 20.000 token di platform Uniswap dan menemukan bahwa lebih dari 50% memiliki karakteristik *scam* yang dapat diidentifikasi dari kode sumber. Pendekatan ini menggabungkan analisis grafik transaksi dengan *machine learning* klasik, namun belum memanfaatkan representasi mendalam kode sumber.

Jin & Li [9] mengadaptasi CodeBERT untuk pencarian semantik dalam *smart contract* (*semantic search*), menunjukkan efektivitas model ini dalam memahami kode Solidity dengan Top-1 accuracy ~78.6%. Namun aplikasinya terbatas pada pencarian dan belum menyentuh klasifikasi multi-label.

Zhang et al. [10] menggabungkan embedding kode sumber dengan *data flow graph* untuk mendeteksi kerentanan *smart contract*, mencapai F1 ~91.3% pada tugas satu kerentanan. Pendekatan ini menginspirasi penggunaan GraphCodeBERT yang secara inheren mengintegrasikan informasi *data flow* dalam representasinya.

Bu et al. [11] mengembangkan SmartBugBERT yang mengadaptasi arsitektur BERT untuk analisis bytecode *smart contract*, mencapai F1-*score* hingga 91% pada deteksi kerentanan tunggal. Penelitian ini membuktikan viabilitas arsitektur BERT untuk keamanan *smart contract*, namun terbatas pada satu jenis kerentanan dan berbasis bytecode, bukan kode sumber.

Model Transformer: BERT, RoBERTa, CodeBERT, dan GraphCodeBERT

Fondasi dari seluruh model yang digunakan dalam penelitian ini adalah arsitektur **BERT** (*Bidirectional Encoder Representations from Transformers*) yang diperkenalkan oleh Devlin et al. [4]. BERT

memanfaatkan mekanisme *self-attention* dua arah (*bidirectional*) untuk membangun representasi kontekstual suatu token berdasarkan seluruh token di kiri maupun kanannya secara bersamaan berbeda dengan model sekuensial sebelumnya seperti LSTM yang memproses teks secara satu arah. Pre-training BERT dilakukan pada dua tugas: *Masked Language Modeling* (MLM), di mana sebagian token disembunyikan dan model belajar memprediksinya; dan *Next Sentence Prediction* (NSP), di mana model belajar mengenali hubungan antar kalimat. Arsitektur standar BERT-base memiliki 12 lapisan *Transformer*, *hidden size* 768, 12 *attention heads*, dan sekitar 110 juta parameter.

RoBERTa

(*Robustly Optimized BERT Pretraining Approach*) [7] merupakan penyempurnaan BERT oleh Liu et al. yang mempertahankan arsitektur yang sama namun mengoptimalkan prosedur pre-training secara menyeluruh. RoBERTa dilatih lebih lama dengan data yang jauh lebih besar (160 GB vs. 16 GB pada BERT), menggunakan *batch size* yang lebih besar, dan menghapus tugas NSP yang terbukti tidak memberi manfaat signifikan. Selain itu, RoBERTa menerapkan *dynamic masking* pola masker diubah setiap kali data digunakan, sehingga model tidak menghafal pola masker yang sama. Hasilnya, RoBERTa secara konsisten melampaui BERT pada berbagai *benchmark* NLP. Meskipun tidak dirancang khusus untuk kode, kemampuan *transfer learning*-nya yang kuat menjadikannya *baseline* yang relevan dan kompetitif dalam penelitian ini.

CodeBERT

Dikembangkan oleh Feng et al. sebagai model pre-trained bimodal yang dilatih pada pasangan teks alami (*natural language*) dan kode (*programming language*) secara bersamaan. Korpus pre-training mencakup lebih dari 6,4 juta pasangan teks-kode dari enam bahasa pemrograman (Python, Java, JavaScript, PHP, Ruby, dan Go) yang diperoleh dari GitHub. CodeBERT menggunakan dua objektif pre-training: MLM atas kedua modalitas, dan *Replaced Token Detection* (RTD) tugas diskriminatif yang membedakan token asli dari token yang diganti oleh generator. Dengan arsitektur identik BERT-base (12 lapisan, 768 *hidden size*, 125 juta parameter), CodeBERT mampu membangun representasi yang menjembatani semantik teks dan struktur kode sumber. Kemampuan ini menjadikannya kandidat yang kuat untuk memahami kode Solidity yang memiliki kemiripan sintaksis dengan bahasa-bahasa yang termasuk dalam korpus pre-training-nya.

GraphCodeBERT

Dikembangkan oleh Guo et al. sebagai evolusi lanjutan CodeBERT yang secara eksplisit memanfaatkan **Data Flow Graph** (DFG) kode sumber selama pra-pelatihan. DFG merepresentasikan ketergantungan data antar variabel: dari mana suatu variabel mendapat nilainya dan variabel mana yang bergantung padanya. Dengan menyertakan informasi struktural ini, GraphCodeBERT memiliki tiga objektif pre-training: MLM (serupa BERT/CodeBERT), *Edge Prediction* (memprediksi apakah ada aliran data antara dua node), dan *Node Alignment* (menyelaraskan token kode dengan node DFG yang bersesuaian). Hasilnya, GraphCodeBERT tidak hanya memahami teks kode secara sekuensial, tetapi juga mengenali hubungan logis dan aliran eksekusi yang tersembunyi di balik kode. Dalam konteks deteksi *scam* pada *smart contract*, kemampuan ini sangat relevan karena banyak pola berbahaya seperti mekanisme Blacklist, manipulasi saldo, atau *hidden ownership* melibatkan aliran data yang tidak lazim yang sulit dideteksi dari urutan token semata.

Ketiga model (RoBERTa, CodeBERT, GraphCodeBERT) berbagi

arsitektur dasar yang sama: 12 lapisan *Transformer encoder*, *hidden size* 768, 12 *attention heads*, dan ~125 juta parameter total. Kesamaan ini memastikan bahwa perbedaan performa yang teramati dalam penelitian ini semata-mata mencerminkan perbedaan pengetahuan yang tersandi selama pre-training, bukan perbedaan kapasitas model.

Taksonomi Scam pada Token ERC-20

Token ERC-20 yang bersifat *scam* tidak selalu memiliki satu mekanisme penipuan tunggal sebuah kontrak dapat secara bersamaan menyembunyikan beberapa mekanisme berbahaya yang diaktifkan pada kondisi berbeda. Penelitian ini mengadopsi taksonomi tujuh label yang mencakup spektrum penipuan yang paling umum ditemui di ekosistem Ethereum.

Honeypot

Honeypot adalah kontrak yang memungkinkan pengguna membeli token tetapi secara diam-diam memblokir transaksi penjualan. Mekanisme ini biasanya diimplementasikan melalui kondisi tersembunyi dalam fungsi transfer atau transferFrom yang memeriksa apakah penjual adalah alamat tertentu (pemilik kontrak), atau dengan memanipulasi variabel *state* sehingga fungsi selalu revert untuk pengguna biasa. Kerugian ekonomi bagi korban sangat signifikan karena dana terjebak dan tidak dapat ditarik.

High Tax

Hight Tax merujuk pada kontrak yang membebankan *fee* transaksi yang eksesif umumnya di atas 30% dari nilai transaksi yang dikirimkan ke alamat pemilik. Pola ini tersebar luas karena secara teknis "legal" (pengguna secara implisit menyetujui kondisi kontrak), namun menyesatkan karena *fee* tidak diungkapkan secara transparan kepada pengguna sebelum transaksi. Dalam kode Solidity, pola ini teridentifikasi melalui kalkulasi *fee* yang besar dalam fungsi transfer.

Balance Manipulation

Balance manipulation Adalah kategori yang mencakup kontrak yang memanipulasi nilai saldo yang ditampilkan kepada pengguna. Kontrak jenis ini biasanya meng-*override* fungsi balanceOf untuk mengembalikan nilai yang dimanipulasi saldo yang ditampilkan jauh lebih besar dari saldo aktual sehingga menciptakan ilusi kekayaan palsu untuk menarik lebih banyak pembeli. Pada level *data flow*, pola ini terlihat dari ketidaksesuaian antara nilai yang disimpan dalam mapping saldo dan nilai yang dikembalikan fungsi query.

Blacklist

Blacklist Merujuk pada kontrak yang memiliki daftar alamat yang diblokir dari melakukan transaksi. Pemilik kontrak dapat menambahkan alamat pembeli ke daftar hitam setelah mereka membeli token, secara efektif mengunci aset mereka. Mekanisme ini diimplementasikan melalui mapping address => bool yang diperiksa pada setiap transaksi, dengan fungsi addToBlacklist yang hanya dapat dipanggil oleh pemilik (*onlyOwner*). Label ini menjadi yang paling sulit dideteksi dalam penelitian ini karena jumlah sampelnya paling sedikit di dataset.

Hidden Owner

Hidden owner adalah kontrak yang menyembunyikan kepemilikan atau akses administratif di balik lapisan abstraksi tambahan. Berbeda dengan kontrak yang menggunakan Ownable standar secara transparan, kontrak Hidden Owner menyimpan referensi ke alamat pemilik dalam variabel yang tidak terdokumentasi, atau menggunakan pola *proxy* dan *delegatecall* untuk menyembunyikan fungsi administratif dari inspeksi superfisial. Dalam konteks DFG, aliran data dari variabel tersembunyi ke fungsi-fungsi *privileged* menjadi sinyal utama.

Rug Pull

Rug Pull merepresentasikan kontrak yang memberikan pemilik kemampuan untuk menarik seluruh likuiditas atau aset secara sepihak. Mekanisme ini dapat berupa fungsi withdraw, drainFunds, atau variasi serupa yang hanya dapat dipanggil oleh pemilik dan memindahkan saldo kontrak ke alamat eksternal. Kontrak Rug Pull sering dikombinasikan dengan kampanye pemasaran agresif untuk menarik investasi sebelum pemilik "menarik karpet" (*rug pull*).

Unlimited Mint

Unlimited mint adalah kontrak yang memungkinkan pemilik mencetak (*mint*) token dalam jumlah tidak terbatas tanpa batasan yang dikodekan dalam kontrak. Berbeda dengan token ERC-20 yang sehat di mana pasokan total (*total supply*) bersifat tetap atau memiliki batas yang jelas, kontrak Unlimited Mint memberikan fleksibilitas bagi pemilik untuk mendilusi nilai token kapan saja. Pola ini teridentifikasi melalui fungsi mint tanpa validasi batas pasokan maksimum.

Ketujuh kategori ini tidak bersifat *mutually exclusive* satu kontrak dapat sekaligus mengandung mekanisme Honeypot *dan* Blacklist *dan* Hidden Owner, misalnya. Hal inilah yang membuat pendekatan klasifikasi multilabel jauh lebih tepat dibanding klasifikasi multikelas biasa untuk masalah ini, karena multilabel mampu merepresentasikan ko-eksistensi beberapa kategori *scam* secara bersamaan dalam satu kontrak.

Klasifikasi Multi-Label

Berbeda dengan klasifikasi biner atau multikelas, klasifikasi multi-label memungkinkan setiap sampel memiliki lebih dari satu label aktif secara bersamaan [12]. Pada konteks *smart contract*, satu kontrak dapat mengandung mekanisme *honeypot* sekaligus *blacklist* dan *rug pull*. Pendekatan *Binary Relevance* yang digunakan dalam penelitian ini memperlakukan setiap label sebagai masalah klasifikasi biner independen dengan *threshold* 0,5 pada output sigmoid.

METODE PENELITIAN

Dataset yang digunakan dalam penelitian ini berasal dari ASSERT-KTH/DISL (Decentralized Intelligence for Smart Contracts Library) [13], yang merupakan koleksi komprehensif smart contract Solidity yang telah di-deploy pada jaringan Ethereum mainnet hingga cutoff date 15 Januari 2024 (sekitar blok 19.010.000). Dataset ini dikurasi oleh tim KTH Royal Institute of Technology dan dipublikasikan melalui paper “DISL: Fueling Research with A Large Dataset of Solidity Smart Contracts” (arXiv:2403.16861). Dataset tersebut tersedia dalam dua subset, yaitu Raw Subset yang berisi 3.298.271 smart contract dengan ukuran 20,5 GB tanpa deduplikasi, serta Decomposed Subset yang terdiri dari 514.506 file Solidity dengan ukuran 2,6 GB yang telah melalui proses deduplikasi menggunakan Jaccard similarity dengan threshold 0,9. Dalam penelitian ini, digunakan Decomposed Subset untuk menghindari bias akibat duplikasi kontrak yang berpotensi menyebabkan kebocoran data (data leakage) antar pembagian data.

Dari total 514.506 kontrak pada Decomposed Subset, dilakukan proses filtering untuk mengidentifikasi kontrak ERC-20 berdasarkan keberadaan fungsi standar, yaitu transfer, transferFrom, approve, allowance, balanceOf, dan totalSupply. Berdasarkan hasil filtering tersebut, dipilih sebanyak 60.000 kontrak ERC-20 menggunakan metode stratified sampling yang mempertimbangkan kompleksitas kode (lines of code) serta periode deployment, sehingga diperoleh representasi dataset yang beragam dan proporsional.

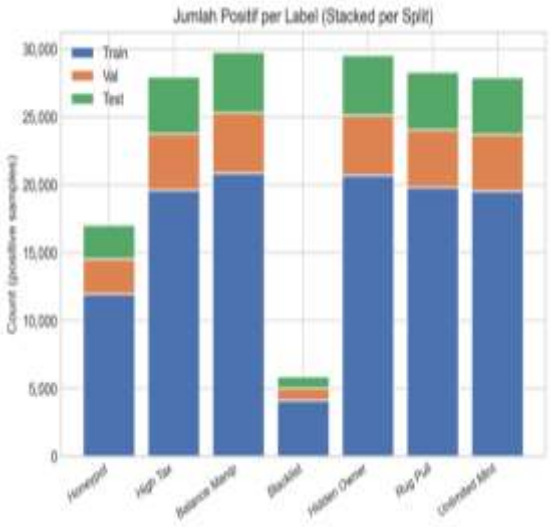
Proses pelabelan dilakukan secara semi-otomatis menggunakan analisis kode statis berbasis aturan (rule-based), yang memanfaatkan pola regex serta analisis AST (Abstract Syntax Tree) pada kode

Solidity untuk mendeteksi tujuh kategori scam, yaitu Honeypot, High Tax, Balance Manipulation, Blacklist, Hidden Owner, Rug Pull, dan Unlimited Mint. Ketujuh kategori ini mencerminkan berbagai modus penipuan yang berbeda, mulai dari yang secara langsung merugikan pengguna seperti Honeypot yang mencegah penjualan token dan High Tax yang mengenakan biaya transaksi berlebih, hingga yang memberikan kontrol penuh kepada pengembang seperti Hidden Owner, Unlimited Mint, dan Rug Pull. Keragaman pola kode pada setiap kategori menyebabkan tugas klasifikasi multilabel menjadi lebih kompleks dibandingkan klasifikasi biner, karena setiap label memiliki karakteristik representasi kode yang berbeda. Dalam konteks ini, label biner (scam vs. legitimate) ditentukan dengan aturan bahwa sebuah kontrak diklasifikasikan sebagai scam apabila memiliki minimal satu dari ketujuh label tersebut.

Setelah proses pelabelan selesai, dataset akhir yang terdiri dari 60.000 kontrak dibagi ke dalam tiga subset menggunakan rasio 70/15/15 untuk data pelatihan, validasi, dan pengujian. Pembagian ini dilakukan secara stratified berdasarkan distribusi label biner untuk memastikan keseimbangan antara kelas scam dan legitimate pada setiap subset. Hasilnya, distribusi kelas biner dipertahankan seimbang dengan rasio 50:50 pada seluruh subset, sehingga model tidak cenderung bias terhadap salah satu kelas, dan metrik evaluasi seperti F1-Score dan Accuracy dapat diinterpretasikan secara lebih objektif.

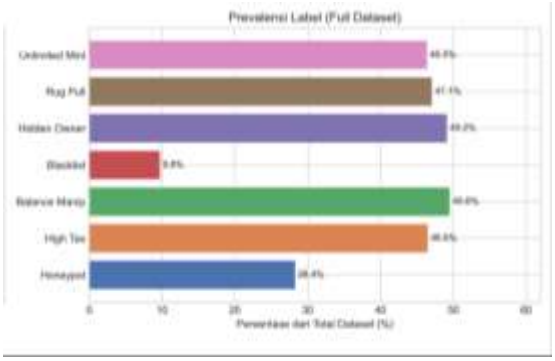
Pada level multilabel, distribusi antar label menunjukkan variasi yang cukup signifikan. Enam label, yaitu High Tax, Balance Manipulation, Hidden Owner, Rug Pull, dan Unlimited Mint, memiliki jumlah sampel positif yang relatif tinggi, berkisar antara 27% hingga 50%, sedangkan label Blacklist memiliki proporsi yang jauh lebih kecil, yaitu sekitar 9,58%. Ketimpangan ini berpotensi mempengaruhi kemampuan model dalam mempelajari pola pada label Blacklist, sehingga menjadi salah satu tantangan utama dalam penelitian ini. Selain itu, analisis statistik terhadap kode sumber menunjukkan adanya variasi panjang kontrak yang sangat besar, dengan rata-rata 22.563 token, median 17.606 token, serta rentang dari 339 hingga 897.115 token. Kondisi ini mengindikasikan bahwa sebagian besar kontrak akan mengalami proses truncation yang signifikan ketika diproses menggunakan batas maksimum panjang sekuens sebesar 256 token.

Distribusi ketujuh label multilabel pada keseluruhan dataset memperlihatkan adanya kesenjangan yang cukup jelas antara label Blacklist dengan label lainnya, sementara label High Tax, Balance Manipulation, Hidden Owner, Rug Pull, dan Unlimited Mint memiliki frekuensi yang relatif seimbang, serta Honeypot berada pada tingkat menengah. Hal ini semakin menegaskan kompleksitas permasalahan multilabel dalam deteksi scam berbasis kode smart contract.



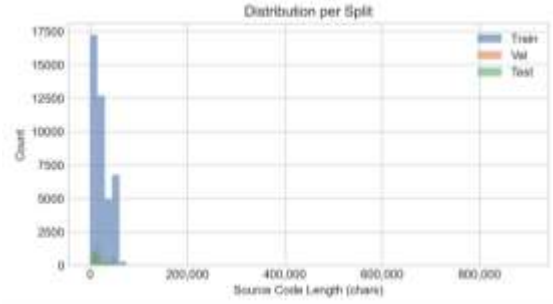
Gambar 1. Distribusi label multilabel pada keseluruhan

Distribusi label multilabel pada keseluruhan dataset (60.000 sampel). Label Blacklist tampak jauh lebih rendah (9,78%) dibandingkan enam label lainnya (27–50%), mengindikasikan potensi tantangan ketidakseimbangan kelas yang signifikan.



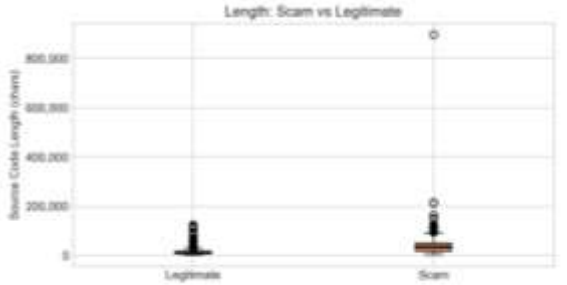
Gambar 2. distribusi panjang kode sumber

Gambar 2 menampilkan distribusi panjang kode sumber dalam satuan karakter sebelum tokenisasi dan truncation. Distribusi yang sangat right-skewed (ekor panjang ke kanan) mengkonfirmasi bahwa sebagian besar kontrak berukuran moderat, namun ada sejumlah kontrak ekstrem yang mencapai ratusan ribu karakter. Kondisi ini menekankan pentingnya strategi truncation yang cermat dan memotivasi eksplorasi model long-context pada penelitian lanjutan.

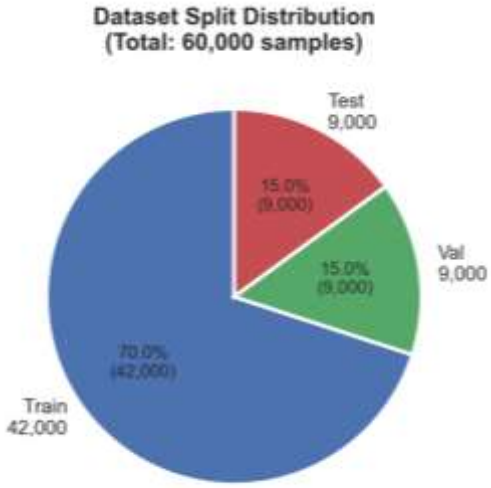


Gambar 3. Distribusi panjang kode sumber (jumlah karakter) sebelum tokenisasi

Gambar 3. Distribusi panjang kode sumber (jumlah karakter) sebelum tokenisasi. Distribusi bersifat right-skewed dengan ekor yang sangat panjang mayoritas kontrak berukuran di bawah 50.000 karakter, namun terdapat outlier yang mencapai lebih dari 800.000 karakter. Membandingkan distribusi label pada ketiga split (train, validation, test) secara bersamaan. Konsistensi distribusi antar split merupakan indikator keberhasilan stratified sampling: ketiga kurva memiliki pola yang hampir identik, memastikan bahwa model tidak menghadapi distribusi yang berbeda saat diuji pada test set dibandingkan saat dilatih di gambarkan pada gambar 4 di bawah ini



Gambar 4. Dstribusi label pada ketiga split



Gambar 5. Perbandingan distribusi label pada train/val/test split

Gambar 4 menampilkan matriks ko-kemunculan (co-occurrence matrix) yang menunjukkan seberapa sering dua label muncul bersamaan dalam satu kontrak. Warna yang lebih gelap menandakan ko-kemunculan yang lebih sering. Pola ini memberikan wawasan penting tentang hubungan semantik antar jenis scam: misalnya, kontrak yang menggunakan mekanisme Rug Pull seringkali juga mengimplementasikan Hidden Owner (karena keduanya bergantung pada kontrol eksklusif pemilik), sementara Blacklist cenderung lebih independen dari kategori lainnya.

Gambar 6. Matriks ko-kemunculan label pada dataset

Gambar 4. Matriks ko-kemunculan label pada dataset. Nilai pada sel (i,j) menunjukkan proporsi sampel yang memiliki kedua label i dan j sekaligus. Ko-kemunculan tinggi antara High Tax–Rug Pull dan Hidden Owner Unlimited Mint mengindikasikan adanya "paket" pola scam yang sering diimplementasikan bersama.

Arsitektur Model

Ketiga model menggunakan arsitektur yang seragam untuk memastikan perbandingan yang adil. Perbedaan utama antar model terletak pada proses pra-pelatihan yang digunakan. CodeBERT dilatih menggunakan kombinasi Masked Language Modeling (MLM) dan Replaced Token Detection (RTD) pada pasangan teks dan kode dari enam bahasa pemrograman, sehingga mampu memahami hubungan antara bahasa alami dan struktur kode. Sementara itu, RoBERTa hanya menggunakan objektif MLM pada teks bahasa alami dalam skala besar tanpa pengetahuan khusus mengenai kode, namun tetap memiliki kemampuan representasi yang kuat melalui transfer learning. Di sisi lain, GraphCodeBERT mengembangkan pendekatan lebih lanjut dengan menggabungkan MLM, Edge Prediction, dan Node Alignment, serta memanfaatkan informasi data flow graph, sehingga model ini mampu menangkap hubungan antar variabel dan aliran eksekusi dalam kode secara lebih mendalam.

```
[Input: Solidity Source Code]
        ↓
[Tokenizer (BPE, max_length=256)]
        ↓
[Pretrained Transformer Encoder]
  - 12 layers
  - Hidden size: 768
  - 12 attention heads
  - ~125M parameters
        ↓
[CLS Token Representation (768-dim)]
        ↓
[Dropout (p=0.1)]
        ↓
[Classification Head]
  - Binary:     Linear(768 → 2) + Softmax
  - Multilabel: Linear(768 → 7) + Sigmoid
        ↓
[Output: Scam Label(s)]
```

Konfigurasi Training

Seluruh eksperimen menggunakan konfigurasi hyperparameter yang identik untuk ketiga model, sehingga perbedaan performa yang teramati dapat dikaitkan semata-mata dengan perbedaan arsitektur pra-pelatihan, bukan perbedaan kondisi pelatihan. Tabel 4 merangkum konfigurasi lengkap yang digunakan. Beberapa keputusan konfigurasi krusial perlu dijelaskan. Pertama, learning rate 5×10^{-6} yang sangat kecil dipilih untuk menghindari catastrophic forgetting pada bobot model yang sudah pre-trained nilai ini lebih konservatif dari learning rate default fine-tuning (2×10^{-5}) karena keterbatasan memori GPU yang memaksa penggunaan batch size kecil (4). Kedua, max sequence length 256 token adalah kompromi antara cakupan kode dan efisiensi memori; penggunaan nilai lebih tinggi (512 atau 1024) tidak memungkinkan dalam lingkungan GPU yang tersedia. Ketiga, strategi early stopping berbasis validation F1 (bukan validation loss) dipilih karena F1 lebih informatif dalam konteks ketidakseimbangan kelas ringan pada tugas multilabel. Keempat, Binary Cross-Entropy per label pada tugas multilabel memperlakukan setiap label sebagai masalah biner independen, yang merupakan implementasi Binary Relevance

yang sederhana namun terbukti efektif pada banyak tugas multi-label. Training dilakukan pada GPU dengan framework PyTorch dan Hugging Face Transformers, dengan model terbaik (best checkpoint) disimpan berdasarkan validation F1-Score tertinggi.

Table 1. Hyperparameter Training (Seragam untuk Semua Model)

Parameter	Nilai
Max Sequence Length	256 token
Batch Size	4
Learning Rate	5×10^{-6}
Optimizer	AdamW
Weight Decay	0,01
Warmup Steps	500
Scheduler	Linear Warmup + Linear Decay
Max Epochs	5
Early Stopping	Patience 3, berdasarkan Val F1
Gradient Clipping	Max Norm 1.0
Loss (Binary)	Cross-Entropy Loss
Loss (Multilabel)	Binary Cross-Entropy per label
Threshold Klasifikasi	0,5 (default)

Metrik Evaluasi

Evaluasi dilakukan pada test set yang terdiri dari 9.000 sampel dengan menggunakan beberapa metrik utama, yaitu accuracy, precision, recall, F1-score, dan AUC-ROC. Accuracy digunakan untuk mengukur proporsi prediksi yang benar terhadap seluruh sampel, sedangkan precision mengukur tingkat ketepatan prediksi positif. Recall menunjukkan kemampuan model dalam mendeteksi seluruh data positif, dan F1-score merupakan harmonic mean dari precision dan recall yang memberikan keseimbangan antara keduanya. Sementara itu, AUC-ROC digunakan untuk mengukur kemampuan model dalam membedakan kelas pada berbagai nilai threshold. Pada skenario multilabel, seluruh metrik dihitung menggunakan pendekatan macro average, yaitu rata-rata tak tertimbang per label, sehingga setiap kategori memiliki kontribusi yang sama tanpa dipengaruhi oleh frekuensi kemunculannya dalam dataset.

PEMBAHASAN

Dinamika Pelatihan

Klasifikasi Biner

Tabel 5 menyajikan rekam jejak metrik pelatihan ketiga model selama 5 epoch pada tugas klasifikasi biner. Kolom *Train Loss* dan *Val F1* dipilih sebagai representasi primer karena keduanya mencerminkan dua perspektif berbeda: *Train Loss* menggambarkan seberapa cepat model menyesuaikan diri dengan data latih, sementara *Val F1* mengukur kemampuan generalisasi model pada data yang belum pernah dilihat selama pelatihan.

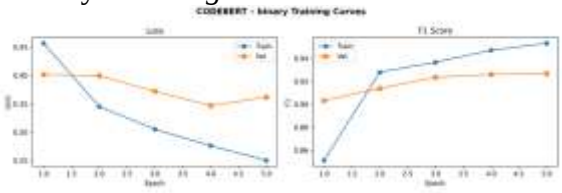
Tabel 5. Kurva Pelatihan Klasifikasi Biner (per Epoch)

Epoch	CodeBERT Train Loss	CodeBERT Val F1	RoBERTa Train Loss	RoBERTa Val F1	GraphCodeBERT Train Loss	GraphCodeBERT Val F1
1	0,4567	0,9033	0,4843	0,9058	0,4246	0,9089
2	0,3451	0,9139	0,3765	0,9138	0,3186	0,9225
3	0,3053	0,9235	0,3321	0,9207	0,2772	0,9230
4	0,2762	0,9260	0,3026	0,9259	0,2481	**0,9289**
5	0,2503	**0,9267**	0,2659	**0,9265**	0,2229	**0,9289**

Beberapa pola penting dapat diamati dari Tabel 5. GraphCodeBERT memulai epoch pertama dengan *Train Loss* terendah (0,4246) dan *Val F1* tertinggi (0,9089), mengindikasikan bahwa inisialisasi berbasis *data flow graph* memberikan titik awal yang lebih menguntungkan. CodeBERT dan RoBERTa memiliki pola

penurunan *loss* yang serupa, tetapi RoBERTa memulai dengan *Train Loss* tertinggi (0,4843) mencerminkan jarak semantik yang lebih besar antara domain teks umum tempat RoBERTa pre-trained dengan domain kode Solidity. Menariknya, selisih *Val F1* akhir (epoch 5) antara ketiga model sangat tipis: hanya 0,0002 poin (GraphCodeBERT 0,9289 vs. RoBERTa 0,9265 vs. CodeBERT 0,9267), menunjukkan konvergensi yang hampir seragam.

Gambar 5 memperlihatkan kurva pelatihan CodeBERT pada tugas klasifikasi biner secara visual. Penurunan *training loss* yang mulus dan peningkatan *validation* F1 yang konsisten mengkonfirmasi stabilitas proses *fine-tuning*. Tidak terlihat gejala *overfitting* yang signifikan karena *validation* F1 terus meningkat seiring turunnya *training loss*.



Gambar 6. Kurva pelatihan CodeBERT pada klasifikasi biner. Grafik atas menampilkan penurunan* training loss *dan* validation loss *per epoch; grafik bawah menampilkan peningkatan* validation F1*. Konvergensi tercapai pada epoch ke-5 dengan Val F1 = 0,9267.

Gambar 6 menampilkan kurva pelatihan RoBERTa. Meskipun model ini dimulai dengan *training loss* yang lebih tinggi dibanding CodeBERT dan GraphCodeBERT konsekuensi dari pre-training pada domain teks umum yang lebih jauh dari kode laju penurunan *loss*-nya konsisten dan *validation* F1 akhirnya tidak kalah dari CodeBERT. Ini mendemonstrasikan kemampuan *transfer learning* RoBERTa yang kuat untuk beradaptasi ke domain baru.

Gambar 7 menunjukkan kurva pelatihan GraphCodeBERT. Model ini memulai dengan *training loss* terendah (0,4246) dan secara konsisten mencatat *validation* F1 tertinggi di setiap epoch, mengkonfirmasi bahwa pengetahuan *data flow* yang tersandi dalam pre-training GraphCodeBERT memberikan keuntungan yang terukur sejak iterasi awal.

Ketiga model menunjukkan konvergensi yang stabil tanpa gejala *overfitting* yang signifikan, dibuktikan oleh nilai *validation* F1 yang terus meningkat seiring turunnya *training loss* di setiap epoch.

Klasifikasi Multilabel

Tabel 6 menyajikan dinamika pelatihan ketiga model pada tugas yang lebih kompleks, yaitu klasifikasi multilabel tujuh kelas. Dibandingkan dengan tugas biner, *training loss* pada epoch pertama (0,35–0,38) secara keseluruhan lebih rendah karena fungsi *loss* yang digunakan berbeda (Binary Cross-Entropy per label vs. Cross-Entropy tunggal), namun *Val F1* awal (0,80–0,82) jauh lebih rendah dibanding pada tugas biner (0,90–0,91) mencerminkan kompleksitas tugas yang meningkat drastis ketika model harus membuat tujuh keputusan biner sekaligus.

Tabel 6. Kurva Pelatihan Klasifikasi Multilabel (per Epoch)

Epoch	CodeBERT Train Loss	CodeBERT Val F1	RoBERTa Train Loss	RoBERTa Val F1	GraphCodeBERT Train Loss	GraphCodeBERT Val F1
1	0,3720	0,8065	0,3812	0,8088	0,3553	0,8234
2	0,2620	0,8328	0,2701	0,8351	0,2529	0,8397
3	0,2311	0,8504	0,2398	0,8528	0,2238	0,8569
4	0,2117	0,8578	0,2210	0,8599	0,2045	0,8606
5	0,1983	**0,8622**	0,2072	**0,8679**	0,1908	**0,8643**

Pola yang paling menarik pada Tabel 6 adalah perubahan urutan peringkat model dibanding tugas biner. Pada epoch pertama, GraphCodeBERT masih memimpin dengan *Val F1* 0,8234, namun pada epoch kelima RoBERTa justru meraih *Val F1* tertinggi (0,8679), mengungguli GraphCodeBERT (0,8643) dan CodeBERT

(0,8622). Fenomena ini menandakan bahwa pada tugas multilabel, kemampuan *recall* yang lebih tinggi dari RoBERTa kecenderungan untuk lebih agresif memprediksi label positif memberikan keuntungan yang terakumulasi seiring bertambahnya epoch. Ketiga model tidak menunjukkan tanda-tanda *overfitting*, yang terlihat dari peningkatan *Val F1* yang konsisten hingga epoch terakhir.

Gambar 8 memperlihatkan kurva pelatihan CodeBERT pada tugas multilabel. Penurunan *training loss* yang lebih curam dibanding tugas biner mencerminkan bahwa model memiliki ruang belajar yang lebih besar ketika menghadapi tujuh label sekaligus. Peningkatan *Val F1* dari 0,8065 ke 0,8622 (delta +0,056) lebih besar dibanding delta pada tugas biner (+0,023), menunjukkan potensi peningkatan yang masih terbuka jika pelatihan dilanjutkan.

Gambar 9 menampilkan kurva pelatihan GraphCodeBERT pada tugas multilabel. Meskipun GraphCodeBERT memulai dengan *Val F1* terbaik di epoch pertama (0,8234 vs. 0,8065–0,8088 model lain), laju peningkatannya sedikit lebih lambat dibanding RoBERTa pada epoch berikutnya. Hal ini mungkin disebabkan karena informasi *data flow graph* lebih relevan untuk membuat keputusan tunggal (biner) dibanding keputusan multi-dimensi yang membutuhkan lebih banyak fleksibilitas representasi semantik teks.

Evaluasi Klasifikasi Biner

Setelah proses pelatihan selesai dan model terbaik dipilih berdasarkan *validation* F1, evaluasi akhir dilakukan pada test set yang terdiri dari 9.000 kontrak (4.500 *scam* dan 4.500 *legitimate*) yang tidak pernah dilihat selama pelatihan maupun proses *model selection*. Tabel 7 menyajikan hasil evaluasi lengkap ketiga model menggunakan lima metrik utama.

Tabel 7. Perbandingan Metrik Evaluasi Klasifikasi Biner (Test Set, n=9.000)

Model	Accuracy	Precision	Recall	F1-Score	AUC
CodeBERT	0,9250	0,9160	0,9358	0,9258	0,9797
RoBERTa	0,9234	0,9220	0,9251	0,9236	0,9796
GraphCodeBERT	**0,9290**	**0,9229**	**0,9362**	**0,9295**	**0,9808**
Selisih Maks–Min	*0,0056*	*0,0069*	*0,0111*	*0,0059*	*0,0012*

Tabel 7 memperlihatkan bahwa GraphCodeBERT menempati posisi teratas pada seluruh lima metrik evaluasi secara konsisten. F1-*Score* 0,9295 berarti model ini benar dalam ~93 dari setiap 100 prediksi *scam* yang dibuat (dengan mempertimbangkan keseimbangan antara *precision* dan *recall*). AUC 0,9808 mengkonfirmasi kemampuan diskriminasi yang luar biasa angka ini berarti jika satu kontrak *scam* dan satu kontrak *legitimate* dipilih secara acak, model akan memberikan skor probabilitas lebih tinggi pada *scam* dalam ~98,1% kasus.

CodeBERT berada di posisi kedua dengan F1-*Score* 0,9258, unggul atas RoBERTa (0,9236) meskipun selisihnya sangat tipis (0,0022). Menariknya, CodeBERT mencatat *Recall* tertinggi kedua (0,9358) yang hampir menyamai GraphCodeBERT (0,9362), menunjukkan bahwa model berbasis kode ini sangat efektif dalam menangkap kontrak *scam* yang sebenarnya. Sebaliknya, RoBERTa menunjukkan *Precision* tertinggi kedua (0,9220) dengan *Recall* yang sedikit lebih rendah, mengindikasikan perilaku yang lebih "konservatif" lebih hati-hati dalam menandai kontrak sebagai *scam*, sehingga lebih jarang membuat kesalahan *false positive* tetapi lebih sering melewatkan *scam* yang sebenarnya.

Gambar 10 memvisualisasikan perbandingan F1-*Score* dan AUC ketiga model dalam bentuk *bar chart* yang memudahkan perbandingan visual. Kesenjangan antar model yang sangat kecil pada kedua metrik ini (F1: hanya 0,006;

AUC: hanya 0,001) tergambar jelas secara grafis, memperkuat kesimpulan bahwa semua model memiliki kemampuan yang setara secara praktis.

Gambar 10. Perbandingan F1-Score dan AUC ketiga model pada klasifikasi biner. Batang yang hampir sama tinggi mengkonfirmasi bahwa perbedaan performa antar model sangat kecil (≤0,006 F1), meskipun GraphCodeBERT secara konsisten berada di posisi teratas pada semua metrik.

GraphCodeBERT menempati posisi teratas pada seluruh metrik. F1-*Score* 0,9295 dan AUC 0,9808 mengkonfirmasi bahwa model ini paling efektif dalam membedakan kontrak *scam* dari kontrak legitim secara keseluruhan. Keunggulan ini konsisten dengan desain arsitekturnya yang secara eksplisit memanfaatkan informasi *data flow* dari kode sumber, sehingga model mampu mengenali pola struktural kode *smart contract* lebih mendalam.

Perlu dicatat bahwa selisih F1-*Score* di antara ketiga model sangat kecil (~0,006 poin), mengindikasikan bahwa representasi kode sumber Solidity yang dikodekan masing-masing model sudah cukup informatif. Nilai AUC >0,97 untuk ketiga model mengkonfirmasi kemampuan diskriminasi yang sangat tinggi.

Confusion Matrix dan ROC Curve Biner

Confusion matrix memberikan gambaran yang lebih granular tentang jenis kesalahan yang dibuat oleh setiap model, melampaui apa yang dapat ditangkap oleh metrik agregat seperti F1-*Score*. Tabel 8 merangkum komponen *confusion matrix* ketiga model beserta turunannya: *False Positive Rate* (FPR) dan *False Negative Rate* (FNR).

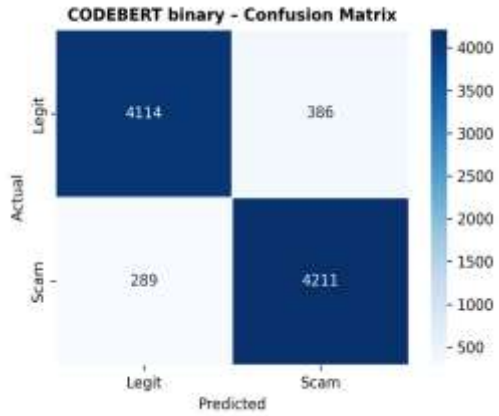
Tabel 8. Confusion Matrix Detail Klasifikasi Biner (Test Set)

Model	TP	TN	FP	FN	FPR	FNR
CodeBERT	4.211	4.114	386	289	8,58%	6,42%
RoBERTa	4.163	4.148	352	337	7,82%	7,49%
GraphCodeBERT	4.213	4.148	352	287	7,82%	6,38%

TP = True Positive (Scam terdeteksi benar), TN = True Negative (Legitimate benar), FP = False Positive (Legitimate salah ditandai Scam), FN = False Negative (Scam lolos tidak terdeteksi)

Dari Tabel 8, dua temuan penting muncul. Pertama, GraphCodeBERT dan RoBERTa berbagi FPR yang identik (7,82%) keduanya salah menandai 352 kontrak *legitimate* sebagai *scam*. Ini menunjukkan bahwa kedua model memiliki ambang keputusan yang setara dalam menghindari *false alarm*. Namun, GraphCodeBERT unggul pada FNR terendah (6,38% vs. 7,49% RoBERTa), artinya GraphCodeBERT melewatkan lebih sedikit kontrak *scam* yang sebenarnya (287 vs. 337 FN). Dalam konteks keamanan *blockchain*, FNR yang lebih rendah lebih kritis karena *false negative* berarti kontrak berbahaya lolos dari sistem deteksi dan berpotensi merugikan pengguna. Kedua, CodeBERT memiliki FPR tertinggi (8,58%) dengan FNR yang kompetitif (6,42%), mencerminkan kecenderungan untuk lebih "agresif" dalam mendeteksi *scam* lebih banyak *false alarm* namun juga lebih sedikit *miss*.

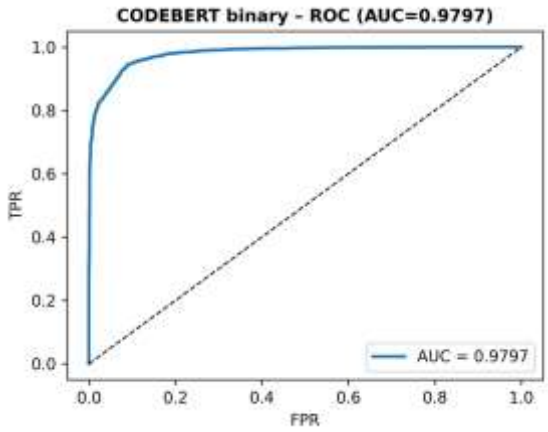
Gambar 11, 12, dan 13 masing-masing menampilkan *confusion matrix* visual CodeBERT, RoBERTa, dan GraphCodeBERT. Warna sel yang lebih gelap pada diagonal utama (TP dan TN) mengkonfirmasi dominasi prediksi yang benar untuk ketiga model. Perbedaan antar model terutama terlihat pada ukuran sel FN (kanan bawah) yang lebih kecil pada GraphCodeBERT dibanding RoBERTa.



Gambar 11. Confusion Matrix CodeBERT pada klasifikasi biner. Diagonal utama (TP=4.211, TN=4.114) mendominasi, dengan FP=386 dan FN=289. FPR sebesar 8,58% adalah yang tertinggi di antara ketiga model, mengindikasikan CodeBERT cenderung lebih agresif dalam menandai kontrak sebagai* scam.

GraphCodeBERT dan RoBERTa memiliki *False Positive Rate* yang sama (7,82%), sementara GraphCodeBERT unggul dengan *False Negative Rate* terendah (6,38%) berarti lebih jarang melewatkan kontrak *scam* yang sebenarnya.

Kurva ROC (*Receiver Operating Characteristic*) pada Gambar 14–16 memberikan perspektif yang berbeda: bukan pada *threshold* 0,5 yang tetap, melainkan pada kemampuan model untuk memisahkan kedua kelas di seluruh rentang *threshold* yang mungkin. Nilai AUC yang mendekati 1 menandakan separabilitas kelas yang hampir sempurna.



Gambar 14. ROC Curve CodeBERT pada klasifikasi biner (AUC = 0,9797). Kurva yang mendekati sudut kiri atas menunjukkan bahwa model mampu mencapai* True Positive Rate *yang sangat tinggi dengan* False Positive Rate *yang rendah di berbagai* threshold. *AUC 0,9797 mengkonfirmasi kemampuan diskriminasi yang sangat baik.

Evaluasi Klasifikasi Multilabel

Pada skenario klasifikasi multilabel, setiap model dievaluasi kemampuannya mendeteksi tujuh kategori *scam* secara simultan dari kode sumber yang sama. Evaluasi ini secara inheren lebih menantang: satu kesalahan prediksi pada salah satu dari tujuh label sudah cukup untuk menurunkan metrik Macro Average, sehingga model yang baik harus konsisten di semua kategori, termasuk kategori yang jumlah sampelnya terbatas. Tabel 9 merangkum perbandingan metrik rata-rata (*macro-averaged*) ketiga model.

Tabel 9. Perbandingan Metrik Rata-Rata (Macro) Klasifikasi Multilabel (Test Set)

Model	Accuracy	Precision	Recall	F1-Score	AUC
CodeBERT	0,8198	0,8806	0,8485	0,8625	0,9656
RoBERTa	0,8188	0,8791	**0,8608**	**0,8681**	0,9670
GraphCodeBERT	0,8197	**0,8822**	0,8556	0,8670	**0,9672**
Selisih Maks–Min	*0,0010*	*0,0031*	*0,0123*	*0,0056*	*0,0016*

Hasil pada Tabel 9 memperlihatkan pergeseran yang signifikan dibanding klasifikasi biner. Jika pada tugas biner GraphCodeBERT mendominasi secara absolut, pada tugas multilabel persaingan menjadi jauh lebih ketat. RoBERTa memimpin pada F1-*Score* (0,8681) dan Recall (0,8608), sementara GraphCodeBERT unggul tipis pada Precision (0,8822) dan AUC (0,9672).

Selisih F1 antara RoBERTa dan GraphCodeBERT hanya 0,0011 poin perbedaan yang dalam praktik tidak signifikan secara statistik. CodeBERT konsisten di posisi ketiga namun tidak jauh tertinggal (F1 = 0,8625, selisih 0,0056 dari yang terbaik).

Penurunan Accuracy dari ~0,92 (biner) ke ~0,82 (multilabel) adalah wajar dan mencerminkan peningkatan kompleksitas tugas: Accuracy multilabel dihitung per-sampel menggunakan *exact match ratio* atau *Hamming Accuracy*, di mana satu kesalahan label saja menyebabkan sampel tersebut dianggap salah sepenuhnya.

Gambar 17 memvisualisasikan perbandingan kelima metrik untuk ketiga model secara bersamaan. Pola batang yang hampir sama tinggi untuk semua metrik kecuali Recall mengkonfirmasi bahwa perbedaan utama antar model terletak pada *trade-off* antara Precision dan Recall: GraphCodeBERT lebih "presisi" (lebih sedikit *false positive* per label), sementara RoBERTa lebih "sensitif" (lebih sedikit *false negative* per label).

Gambar 17. Perbandingan metrik rata-rata (macro) ketiga model pada klasifikasi multilabel. Perhatikan bahwa RoBERTa unggul pada F1 dan Recall, sementara GraphCodeBERT unggul pada Precision dan AUC mencerminkan trade-off yang berbeda antara false positive dan false negative antar model.

Berbeda dengan hasil klasifikasi biner, tidak ada model yang mendominasi secara absolut pada tugas multilabel. RoBERTa memimpin pada F1-*Score* (0,8681) dan Recall (0,8608), sementara GraphCodeBERT unggul pada Precision (0,8822) dan AUC (0,9672). Selisih antara RoBERTa dan GraphCodeBERT hanya 0,0011 F1, sehingga secara praktis keduanya setara.

Fakta bahwa RoBERTa model bahasa teks umum tanpa pengetahuan khusus kode mampu menandingi bahkan melampaui model *code-specific* pada tugas multilabel merupakan temuan yang signifikan. Hal ini menunjukkan bahwa representasi teks kode Solidity (tanpa informasi struktural khusus) sudah cukup informatif untuk tugas multi-label ini.

Perbandingan F1-Score per Label

Analisis per-label memberikan perspektif yang jauh lebih kaya dibanding metrik rata-rata. Tabel 10 merinci F1-*Score* masing-masing model untuk setiap dari tujuh label, memungkinkan identifikasi kekuatan dan kelemahan spesifik tiap model pada setiap kategori *scam*.

Tabel 10. Perbandingan F1-Score per Label Klasifikasi Multilabel

Label	CodeBERT	RoBERTa	GraphCodeBERT	Pemenang
Honeypot	0,8540	0,8555	**0,8572**	GraphCodeBERT
High Tax	0,9120	**0,9140**	0,9128	RoBERTa
Balance Manip	0,9201	0,9229	**0,9233**	GraphCodeBERT
Blacklist	0,6128	**0,6385**	0,6300	RoBERTa
Hidden Owner	0,9210	**0,9241**	0,9237	RoBERTa
Rug Pull	0,9137	**0,9155**	0,9137	RoBERTa
Unlimited Mint	0,9037	0,9061	**0,9080**	GraphCodeBERT
Rata-rata	**0,8625**	**0,8681**	**0,8670**	

Tabel 10 menunjukkan persaingan yang sangat ketat di semua label kecuali Blacklist. Lima dari tujuh label memiliki selisih F1 antar model tidak lebih dari 0,003 poin, mencerminkan bahwa ketiga model belajar pola kode yang serupa untuk kategori-kategori tersebut. RoBERTa unggul pada 4 dari 7 label (High Tax, Blacklist, Hidden Owner, Rug Pull), sementara GraphCodeBERT unggul pada 3

label (Honeypot, Balance Manip, Unlimited Mint). CodeBERT tidak memenangkan satu label pun secara F1, namun selisihnya dari yang terbaik sangat kecil (rata-rata <0,01).

AUC per label pada Tabel 11 memberikan gambaran komplementer tentang kemampuan diskriminasi probabilistik masing-masing model, terlepas dari *threshold* 0,5 yang digunakan.

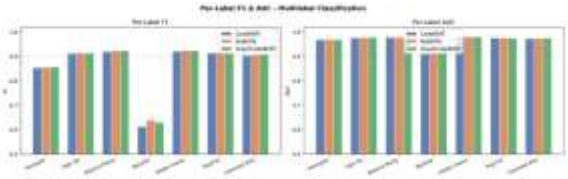
Perbandingan AUC per Label

Tabel 11. Perbandingan AUC per Label Klasifikasi Multilabel

Label	CodeBERT	RoBERTa	GraphCodeBERT	Pemenang
Honeypot	0,9681	**0,9694**	0,9682	RoBERTa
High Tax	0,9763	0,9772	**0,9778**	GraphCodeBERT
Balance Manip	0,9774	0,9781	**0,9782**	GraphCodeBERT
Blacklist	0,9102	0,9151	**0,9178**	GraphCodeBERT
Hidden Owner	0,9784	**0,9796**	0,9795	RoBERTa
Rug Pull	**0,9757**	0,9755	0,9745	CodeBERT
Unlimited Mint	0,9728	0,9742	**0,9746**	GraphCodeBERT
Rata-rata	**0,9656**	**0,9670**	**0,9672**	

Pola yang menarik pada Tabel 11 adalah bahwa GraphCodeBERT mendominasi AUC per-label (unggul di 4 dari 7 label: High Tax, Balance Manip, Blacklist, Unlimited Mint), bahkan untuk label Blacklist yang notabene paling tidak seimbang. AUC Blacklist yang paling tinggi untuk GraphCodeBERT (0,9178) mengkonfirmasi bahwa model ini memiliki kemampuan separasi probabilistik terbaik untuk kontrak Blacklist, meskipun FNR-nya tinggi pada *threshold* 0,5. CodeBERT bahkan memenangkan satu label AUC (Rug Pull: 0,9757), menunjukkan bahwa model ini memiliki kelebihan spesifik dalam mengenali pola *rug pull* secara probabilistik.

Gambar 18 menampilkan *bar chart* grouped yang membandingkan F1-*Score* ketiga model pada setiap label secara berdampingan. Dominasi visual batang-batang yang hampir sama tinggi pada lima label (High Tax hingga Unlimited Mint) versus Blacklist yang jauh lebih pendek menegaskan tantangan khusus label minoritas ini secara intuitif.



Gambar 18. Perbandingan F1-Score per label ketiga model dalam bentuk* bar chart grouped. *Perbedaan mencolok antara Blacklist (F1 ≤0,64) dengan enam label lainnya (F1 ≥0,85) langsung terlihat. Persaingan antar model pada setiap label sangat ketat batang-batang dalam satu grup hampir sama tinggi.

Gambar 19 menyajikan *radar chart* yang memungkinkan perbandingan profil keseluruhan ketiga model secara serentak. Bentuk poligon yang hampir identik untuk ketiga model mengkonfirmasi kesetaraan performa mereka, dengan area yang lebih kecil pada dimensi Blacklist untuk semua model mencerminkan kelemahan bersama yang tidak dapat diatasi oleh arsitektur pre-training manapun tanpa intervensi pada level data.

Gambar 19. Radar chart perbandingan profil performa ketiga model pada 7 label. Tiga poligon yang hampir berimpit mengkonfirmasi kesetaraan performa. Lekukan ke dalam yang dalam pada dimensi Blacklist terlihat pada semua model, mencerminkan tantangan yang bersifat sistemik akibat ketidakseimbangan data, bukan keterbatasan arsitektur model.

Analisis Per-Label Detail

Label dengan Performa Tinggi (F1 > 0,90)

Lima dari tujuh label mencapai F1-*Score* >0,90 secara konsisten di semua model: **High Tax**, **Balance Manipulation**, **Hidden Owner**, **Rug Pull**, dan **Unlimited Mint**. Pola kode untuk kelima kategori ini relatif konsisten dan mudah diidentifikasi:

High Tax: Pola aritmatika fee yang berulang dalam fungsi transfer/transferFrom.; *Balance Manipulation*: Penggantian atau modifikasi balanceOf dengan mapping internal.; *Hidden Owner*: Variabel kepemilikan yang menggunakan nama menyesatkan atau penyimpanan slot tidak konvensional.; *Rug Pull*: Kehadiran fungsi withdraw/drain tanpa pembatasan timelock atau multisig.; serta *Unlimited Mint*: Fungsi mint yang hanya dibatasi oleh modifier onlyOwner.
Kompetisi antar model pada kelima label ini sangat ketat, dengan selisih F1 ≤0,003 poin.

Label Honeypot (F1 ≈ 0,854–0,857)

Label Honeypot menghasilkan F1-*Score* yang lebih rendah (~0,854–0,857) dibanding label lain, meski masih berada pada level baik. Karakteristik *honeypot* yang beragam dan tersembunyi pola pelarangan jual token yang diimplementasikan melalui mekanisme kode yang sangat bervariasi (kondisi tersembunyi, override selektif, state variable manipulasi) menyebabkan model lebih sulit mempelajari representasi yang konsisten. GraphCodeBERT sedikit unggul (F1 = 0,8572), kemungkinan karena pemahaman *data flow*-nya membantu mengenali aliran kontrol yang tersembunyi.

Label Blacklist Tantangan Terbesar (F1 ≈ 0,61–0,64)

Tabel 12. Detail Metrik Label Blacklist

Model	Precision	Recall	F1-Score	AUC
CodeBERT	0,7217	0,5325	0,6128	0,9102
RoBERTa	0,7419	**0,5603**	**0,6385**	0,9151
GraphCodeBERT	0,7353	0,5510	0,6300	**0,9178**

Tabel 12 mengungkapkan kontradiksi yang informatif pada label Blacklist: meskipun AUC masih berada di level 0,91–0,92 (menunjukkan kemampuan separasi probabilistik yang baik), nilai Recall hanya 0,53–0,56, artinya hampir separuh kontrak Blacklist yang sebenarnya ada di test set lolos tanpa terdeteksi. Penyebabnya adalah *threshold* biner 0,5: ketika probabilitas positif prior hanya ~9,6%, model cenderung memberikan skor di bawah 0,5 meskipun probabilitas aktual sudah lebih tinggi dari baseline. RoBERTa meraih F1 terbaik (0,6385) karena Recall-nya sedikit lebih tinggi (0,5603) konsisten dengan kecenderungan model ini yang lebih agresif memprediksi label positif. GraphCodeBERT unggul pada AUC (0,9178), mengkonfirmasi kemampuan diskriminasi probabilistik terbaiknya meskipun pada *threshold* tetap 0,5 hasilnya kurang optimal.

Gambar 20–22 menampilkan *confusion matrix* label Blacklist untuk ketiga model secara berurutan. Pola yang konsisten terlihat pada ketiganya: sel *False Negative* (kanan bawah, kontrak Blacklist yang tidak terdeteksi) secara visual mendominasi dibanding sel *True Positive* (kiri atas, Blacklist yang terdeteksi benar) tanda visual yang jelas dari masalah *class imbalance*.

Gambar 23 menampilkan ROC Curve khusus label Blacklist dari CodeBERT. Meskipun F1-*Score* sangat rendah pada *threshold* 0,5, kurva ROC yang masih jauh

di atas diagonal acak (AUC = 0,9102) membuktikan bahwa model secara fundamental memiliki kemampuan untuk membedakan Blacklist tantangannya murni pada pemilihan *threshold* yang tidak optimal untuk distribusi tidak seimbang ini.

Visualisasi ROC Curve Multilabel Per Model

Meskipun F1-*Score* bervariasi antar label, nilai AUC yang konsisten tinggi (>0,96) untuk seluruh label dan model menunjukkan bahwa model-model ini mampu memberikan skor probabilistik yang baik untuk semua kategori. Bagian ini menyajikan ROC curve individu per label untuk dua model terbaik: CodeBERT dan GraphCodeBERT, sebagai representasi dari model berbasis kode dengan karakteristik yang berbeda.

Untuk setiap kurva, perhatikan dua aspek utama: (1) seberapa cepat kurva "melengkung" ke sudut kiri atas (semakin cepat = semakin baik), dan (2) nilai AUC yang tertera di legenda. Label-label dengan AUC >0,97 (High Tax, Balance Manip, Hidden Owner, Rug Pull) menghasilkan kurva yang hampir vertikal dari titik origin, menandakan separabilitas yang sangat tinggi. Label Blacklist, sebaliknya, menghasilkan kurva yang lebih gradual namun masih jauh di atas diagonal acak.

ROC Curve per Label CodeBERT:

Kedua belas kurva ROC berikut (6 label × 2 model) disajikan secara individual untuk memudahkan inspeksi visual mendalam. Pada CodeBERT, label dengan AUC tertinggi (Hidden Owner: 0,9784 dan Balance Manip: 0,9774) menunjukkan bahwa pola kode untuk kategori ini memiliki fitur yang paling konsisten dan dapat dikenali oleh model.

ROC Curve per Label GraphCodeBERT:

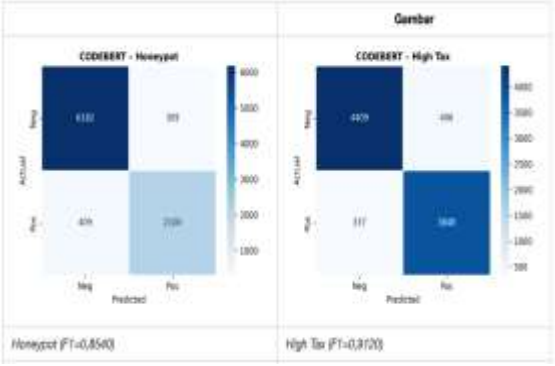
Pada GraphCodeBERT, pola AUC yang konsisten lebih tinggi pada label-label terstruktur (High Tax: 0,9778, Balance Manip: 0,9782, Unlimited Mint: 0,9746) dibanding CodeBERT mencerminkan keunggulan informasi *data flow* yang dimanfaatkan oleh GraphCodeBERT. Secara khusus, GraphCodeBERT memenangkan AUC untuk Blacklist (0,9178 vs. 0,9102 CodeBERT) label yang paling tidak seimbang mengkonfirmasi bahwa representasi *data flow* membantu model memahami pola kontrol aliran kode yang kompleks dalam mekanisme Blacklist.

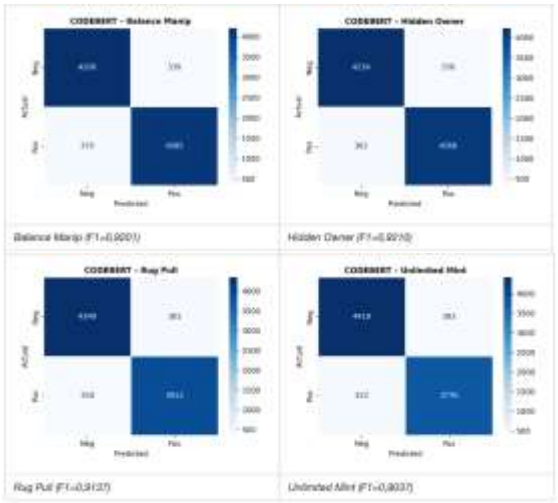
Confusion Matrix Multilabel Semua Label

Visualisasi *confusion matrix* per label memberikan gambaran kuantitatif yang intuitif tentang distribusi prediksi benar dan salah untuk setiap kategori. Pada grid di bawah ini, setiap sel menampilkan nilai mentah (jumlah sampel) agar mudah dibandingkan secara langsung. Pola yang konsisten terlihat: diagonal utama (TP dan TN) mendominasi untuk semua label kecuali Blacklist, di mana sel FN lebih besar dari sel TP konfirmasi visual yang kuat atas tantangan ketidakseimbangan kelas yang telah dianalisis sebelumnya.

CodeBERT Confusion Matrix Multilabel (6 Label Utama):

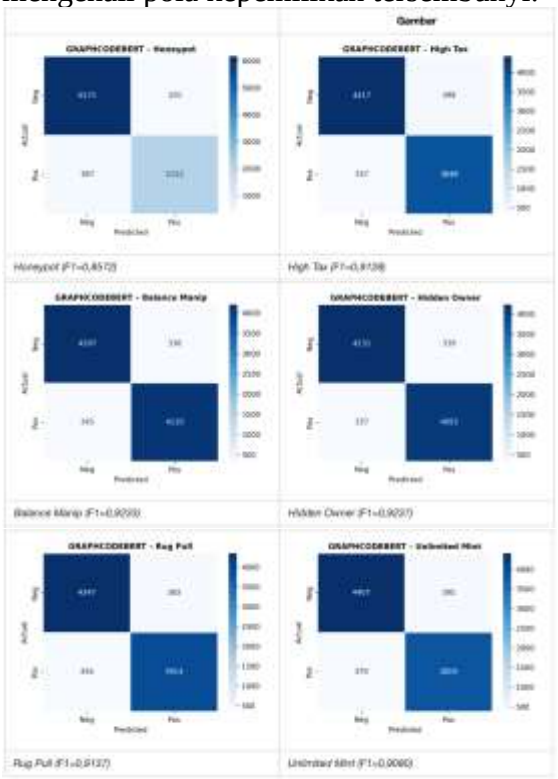
Pada CodeBERT, label-label dengan F1 tinggi (Balance Manip 0,9201, Hidden Owner 0,9210) memiliki distribusi sel yang sangat dominan pada diagonal, dengan FP dan FN yang relatif kecil. Label Honeypot (F1=0,8540) menunjukkan FN yang sedikit lebih banyak dibanding label lain, menandakan bahwa CodeBERT lebih sering melewatkan kontrak Honeypot dibanding kategori lainnya.





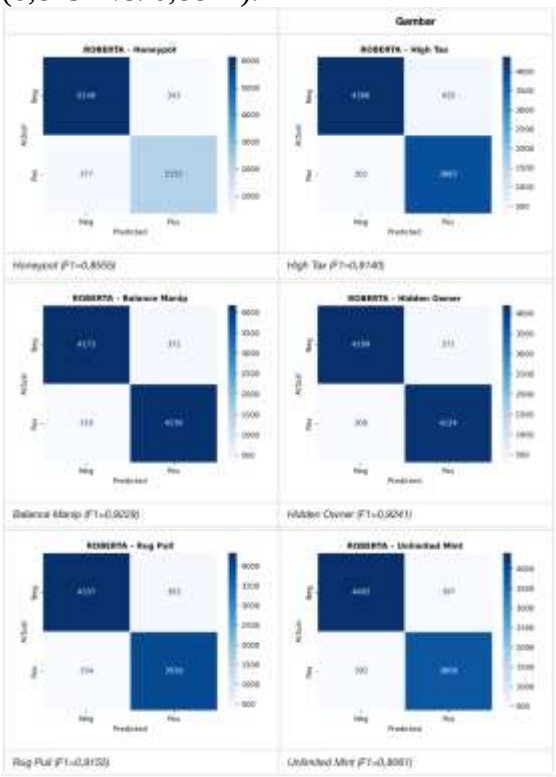
GraphCodeBERT Confusion Matrix Multilabel (6 Label Utama):

Pada GraphCodeBERT, perhatikan bahwa nilai TP pada Balance Manip dan Unlimited Mint sedikit lebih tinggi dibanding CodeBERT, konsisten dengan F1-*Score*-nya yang lebih baik untuk kedua label tersebut. Pola *confusion matrix* GraphCodeBERT untuk Hidden Owner hampir identik dengan RoBERTa, mencerminkan kemampuan setara dalam mengenali pola kepemilikan tersembunyi.



RoBERTa Confusion Matrix Multilabel (6 Label Utama):

Pada RoBERTa, pola yang paling menonjol adalah sel TP yang konsisten lebih tinggi dibanding FN pada hampir semua label mencerminkan kecenderungan RoBERTa yang lebih agresif dalam memprediksi label positif. Hal ini menghasilkan Recall tertinggi secara macro (0,8608) namun juga Precision yang sedikit lebih rendah dibanding GraphCodeBERT (0,8791 vs. 0,8822).



Secara keseluruhan, inspeksi visual seluruh *confusion matrix* multilabel mengkonfirmasi konsistensi hasil kuantitatif yang telah disajikan pada Tabel 10–11: ketiga model memiliki kekuatan deteksi yang setara pada lima label utama, dengan kelemahan bersama pada Blacklist yang bersumber dari ketidakseimbangan data, bukan dari kegagalan model.

Ringkasan Perbandingan Keseluruhan

Sebelum menyimpulkan hasil eksperimen, Tabel 13 merangkum distribusi "kemenangan" setiap model di seluruh skenario dan metrik yang diuji. Ringkasan ini memudahkan pembaca dalam mengidentifikasi pola keunggulan yang tidak selalu terlihat dari metrik individual.

Tabel 13. Rekap Hasil dan Pemenang per Skenario

Skenario	Metrik	Pemenang	Nilai
Biner	F1-Score	**GraphCodeBERT**	0,9295
Biner	AUC	**GraphCodeBERT**	0,9808
Biner	Accuracy	**GraphCodeBERT**	0,9290
Multilabel	F1-Score (Macro)	**RoBERTa**	0,8681
Multilabel	AUC (Macro)	**GraphCodeBERT**	0,9672
Multilabel	Precision	**GraphCodeBERT**	0,8822
Multilabel	Recall	**RoBERTa**	0,8608
Multilabel per-label F1	Win count	**RoBERTa**	4/7 label
Multilabel per-label AUC	Win count	**GraphCodeBERT**	4/7 label

Membaca Tabel 13 secara holistik, terdapat pola yang konsisten dan bermakna. GraphCodeBERT mendominasi pada tugas biner dengan margin yang jelas: menang pada semua tiga metrik biner (F1, AUC, Accuracy). Keunggulan ini merupakan bukti konkret bahwa pengetahuan *data flow* yang dikodekan dalam pre-training GraphCodeBERT memberikan manfaat yang terukur ketika model diminta membuat satu keputusan global tentang karakter keseluruhan kontrak.

Pada tugas multilabel, pola kepemimpinan terbagi: RoBERTa memimpin pada metrik yang berorientasi *recall* (F1-Score Macro dan Recall, serta win count per-label F1 sebesar 4/7), sementara GraphCodeBERT masih memimpin pada metrik yang berorientasi kemampuan diskriminasi (AUC Macro dan win count per-label AUC sebesar 4/7). Pembagian ini bukan kebetulan ia mencerminkan perbedaan mendasar dalam cara kedua model menginternalisasikan pengetahuan kode: GraphCodeBERT memiliki lebih banyak "keyakinan" (probabilitas yang lebih terkalibrasi), sementara RoBERTa lebih "responsif" (cenderung memberikan probabilitas positif lebih tinggi).

CodeBERT, meskipun tidak memenangkan metrik manapun secara keseluruhan, mencatat satu kemenangan per-label AUC (Rug Pull: 0,9757) dan selalu berada dekat di posisi kedua atau ketiga dengan selisih yang tidak signifikan secara praktis. Dalam konteks deployment nyata di mana perbedaan 0,006 F1 tidak terasa, ketiga model dapat dianggap setara dan pilihan antara ketiganya dapat didasarkan pada faktor praktis seperti ukuran model, latency inference, dan ketersediaan infrastruktur.

Efektivitas Model Berbasis Kode vs. Model Teks Umum

Temuan penelitian ini menunjukkan bahwa model *code-specific* (CodeBERT dan GraphCodeBERT) memiliki keunggulan yang konsisten namun marginal (~0,006 F1) dibandingkan model teks umum (RoBERTa) pada klasifikasi biner. Pada klasifikasi multilabel, keunggulan ini bahkan berbalik untuk F1-*Score*, di mana RoBERTa justru unggul. Hal ini mengindikasikan bahwa:

Representasi teks kode Solidity sudah cukup diskriminatif

Untuk membedakan pola *scam* dari kontrak legitim, bahkan tanpa pengetahuan sintaksis kode yang khusus.; **Informasi* data flow *dari GraphCodeBERT memberikan keuntungan lebih nyata pada kasus biner di mana satu keputusan global diperlukan, dibandingkan pada kasus multilabel di mana tujuh keputusan independen harus dibuat sekaligus.; serta RoBERTa memiliki* recall *multilabel yang lebih tinggi karena cenderung lebih "agresif" dalam memprediksi label positif, cocok untuk skenario di mana* False

Negative *lebih berbahaya daripada* False Positive*.

Tantangan Ketidakseimbangan Kelas

Label Blacklist secara konsisten menjadi yang terlemah di semua model (F1 = 0,61–0,64), padahal AUC-nya masih tinggi (0,91–0,92). Ini adalah manifestasi klasik dari masalah ketidakseimbangan kelas dalam klasifikasi multi-label: model memiliki kemampuan diskriminasi yang baik secara probabilistik, tetapi *threshold* biner 0,5 menghasilkan keputusan yang sub-optimal karena prior probabilitas positif hanya ~9,6%. Solusi potensial mencakup:

Threshold optimization

Per label berdasarkan kurva ROC pada validation set.; **Class weighting** pada *loss function* untuk memberikan penalti lebih besar pada *False Negative* Blacklist.; serta **Data augmentation** atau pengumpulan data tambahan khusus Blacklist.

Implikasi Truncation (256 Token)

Dengan rata-rata panjang kode 22.563 token (median 17.606), truncation pada 256 token berarti hanya ~1,1–1,4% kode sumber yang diproses secara utuh. Namun, fakta bahwa ketiga model tetap mencapai F1 >0,92 pada klasifikasi biner menunjukkan bahwa **pola* scam *yang kritis cenderung terletak di bagian awal kode (deklarasi state variables, constructor, fungsi utama). Penggunaan teknik* sliding window *atau* hierarchical encoding* berpotensi meningkatkan performa lebih lanjut.

Keterbatasan Penelitian

Batasan sequence length *: 256 token memotong sebagian besar kode sumber. Kontrak kompleks dengan logika* scam *tersembunyi di fungsi akhir berpotensi terlewat.; Cutoff date dataset: Dataset mencakup kontrak hingga Januari 2024. Teknik* scam *baru yang muncul setelah tanggal tersebut tidak terwakili.; Cakupan jaringan: Penelitian terbatas pada Ethereum ERC-20. Generalisasi ke BNB Smart Chain, Polygon, atau standar lain memerlukan validasi terpisah.; serta Evaluasi real-time: Tidak dilakukan pengujian* deployment *ke jaringan* blockchain secara langsung.

PENUTUP

Kesimpulan

Penelitian ini berhasil mengembangkan dan membandingkan tiga model Transformer CodeBERT, RoBERTa, dan GraphCodeBERT untuk mendeteksi token *scam* ERC-20 melalui dua skenario klasifikasi berbasis kode sumber Solidity. Kesimpulan utama. GraphCodeBERT merupakan model terbaik secara keseluruhan dengan keunggulan jelas pada klasifikasi biner (F1 = 0,9295, AUC = 0,9808). Kemampuan model ini memanfaatkan informasi *data flow* terbukti memberikan keuntungan struktural untuk memahami kode *smart contract*..

Pendekatan multilabel berhasil mendeteksi tujuh kategori scam secara simultan dengan performa memuaskan pada lima dari tujuh label (F1 >0,90). Tidak ada model tunggal yang dominan: RoBERTa unggul pada F1 (0,8681) dan GraphCodeBERT unggul pada AUC (0,9672). Label Blacklist menjadi tantangan bersama (F1 = 0,61–0,64) akibat ketidakseimbangan kelas yang ekstrem. Penanganan khusus melalui optimasi *threshold* atau penambahan data pelatihan diperlukan. RoBERTa, model teks umum, menunjukkan performa kompetitif yang mendekati model *code-specific*, mengkonfirmasi bahwa representasi teks kode Solidity sudah cukup untuk tugas ini, dan membuka peluang penggunaan model bahasa besar (*LLM*) yang lebih kuat di masa mendatang. Selisih performa yang kecil (~0,006 F1) antar model mengindikasikan bahwa ketiga model layak dipertimbangkan untuk implementasi praktis, dengan pilihan bergantung pada trade-off antara Precision, Recall, dan kompleksitas deployment.

Rekomendasi Penelitian Lanjutan

Optimasi *threshold* per label menggunakan data validasi untuk meningkatkan performa Blacklist.; Implementasi *sliding window* atau Longformer untuk kode sumber dengan panjang ekstrem.; Ekspansi dataset ke jaringan BNB Smart Chain dan Polygon.; Integrasi representasi AST/CFG sebagai fitur tambahan.; serta Pengujian *real-time inference* dan optimasi latency untuk deployment pada sistem audit otomatis.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Program Studi Teknik Informatika Universitas Pamulang atas dukungan akademis selama proses penelitian. Terima kasih juga kepada tim KTH Royal Institute of Technology yang telah mempublikasikan dataset DISL secara terbuka melalui Hugging Face, sehingga penelitian berbasis data ini dapat terlaksana. Seluruh eksperimen dilakukan secara mandiri oleh penulis dengan memanfaatkan sumber daya komputasi yang tersedia.

DAFTAR PUSTAKA

[1] F. A. Bakare et al., "Decentralized Finance (DeFi): Opportunities and Risks," *IEEE Access*, 2024.

[2] C. Zhang, "Token Security Analysis on Ethereum: ERC-20 Smart Contract Vulnerabilities," *arXiv preprint*, 2023.

[3] P. Xia et al., "Trade or Trick? Detecting and Characterizing Scam Tokens on Uniswap Decentralized Exchange," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2021.

[4] Y. Wei et al., "Automated Smart Contract Security Analysis Using Large Language Models," *arXiv preprint*, 2025.

[5] Z. Feng et al., "CodeBERT: A Pre-Trained Model for Programming and Natural Languages," in *Findings of EMNLP 2020*, ACL, 2020.

[6] D. Guo et al., "GraphCodeBERT: Pre-training Code Representations with Data Flow," in *ICLR 2021*, OpenReview, 2021.

[7] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv:1907.11692*, 2019.

[8] L. Chen et al., "Detecting Ponzi Schemes on Ethereum: Towards Healthier Blockchain Technology," in *WWW 2018*, ACM, 2018.

[9] Y. Jin and S. Li, "Smart Contract Semantic Search Based on Fine-tuned CodeBERT," *Wuhan University Journal of Natural Sciences*, 2023.

[10] J. Zhang et al., "Smart Contract Vulnerability Detection Using Code Embeddings and Data Flow Graphs," *arXiv preprint*, 2024.

[11] T. Bu et al., "SmartBugBERT: BERT-Based Smart Contract Vulnerability Detection," *IEEE Transactions on Dependable and Secure Computing*, 2025.

[12] M. L. Zhang and Z. H. Zhou, "A Review on Multi-Label Learning Algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[13] G. Morello, M. Eshghie, S. Bobadilla, and M. Monperrus, "DISL: Fueling Research with A Large Dataset of Solidity Smart Contracts," *arXiv:2403.16861*, 2024.

[14] F. Schär, "Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets," *Federal Reserve Bank of St. Louis Review*, 2021.

[15] S. Jiang et al., "Smart Contract Vulnerability Detection: From Pure Neural Network to Interpretable Graph Feature and Expert Pattern Fusion," in *IJCAI 2023*, 2023.