TESIS

DETEKSI SKEMA PONZI PADA BLOCKCHAIN MENGGUNAKAN ALGORITMA WORD EMBEDDING

Disusun oleh:

Nama : **Rizfi Syarif**
NIM : **20.77.1268**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2022

TESIS

DETEKSI SKEMA PONZI PADA BLOCKCHAIN MENGGUNAKAN ALGORITMA WORD EMBEDDING

PONZI SCHEME DETECTION ON BLOCKCHAIN USING WORD EMBEDDING ALGORITHM

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama : **Rizfi Syarif**
NIM : **20.77.1268**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2022

HALAMAN PENGESAHAN

DETEKSI SKEMA PONZI PADA BLOCKCHAIN MENGGUNAKAN ALGORITMA WORD EMBEDDING

PONZI SCHEME DETECTION ON BLOCKCHAIN USING WORD EMBEDDING ALGORITHM

Dipersiapkan dan Disusun oleh

Rizfi Syarif

20.77.1268

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 1 Agustus 2022

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 1 Agustus 2022
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

DETEKSI SKEMA PONZI PADA BLOCKCHAIN MENGGUNAKAN ALGORITMA WORD EMBEDDING

PONZI SCHEME DETECTION ON BLOCKCHAIN USING WORD EMBEDDING ALGORITHM

Dipersiapkan dan Disusun oleh

Rizfi Syarif

20.77.1268

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 1 Agustus 2022

Pembimbing Utama	**Anggota Tim Penguji**
Dr. Andi Sunyoto, M.Kom	**Prof. Dr. Ema Utami, S.Si., M.Kom.**
NIK. 190302052	**NIK. 190302037**
	Dhani Ariatmanto, M.Kom., Ph.D.
Pembimbing Pendamping	**NIK. 190302197**
Drs. Asro Nasiri, M.Kom.	**Dr. Andi Sunyoto, M.Kom**
NIK. 190302152	**NIK. 190302052**

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 1 Agustus 2022
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : **Rizfi Syarif**
NIM : 20.77.1268
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Deteksi Skema Ponzi Pada Blockchain Menggunakan Algortima Word Embedding

Dosen Pembimbing Utama : Dr. Yuni Sutiyono, M.Kom
Dosen Pembimbing Pendamping : Drs. Asro Nasiri, M.Kom

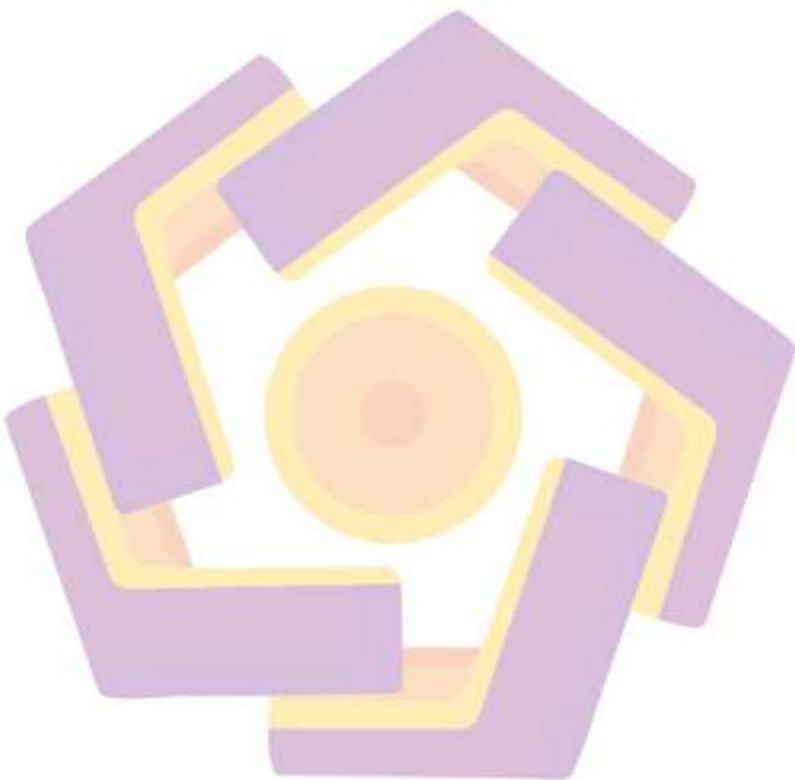
1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 1 Agustus 2022
Yang Menyatakan,

Rizfi Syarif

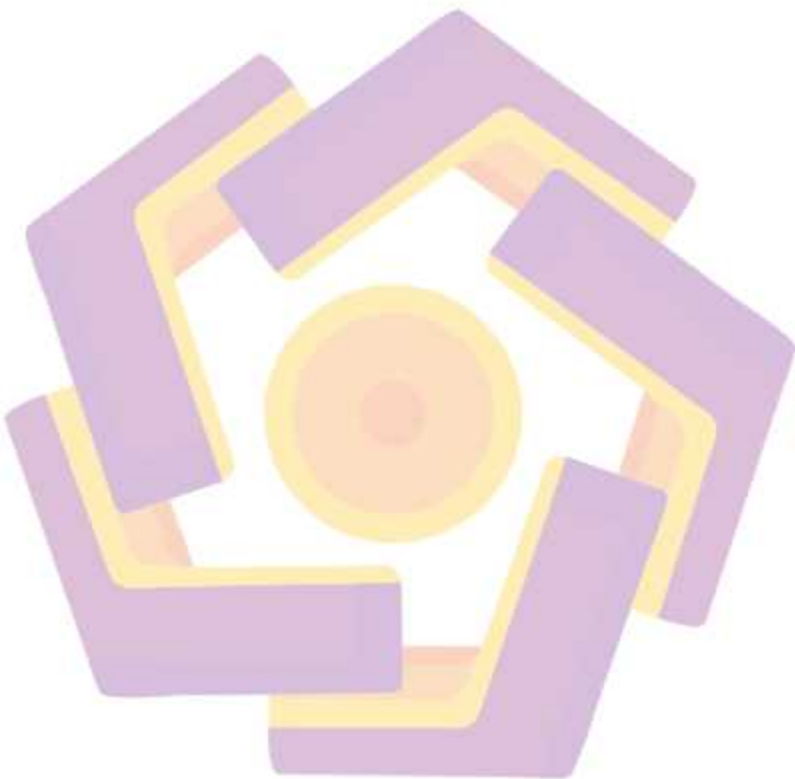
HALAMAN PERSEMBAHAN

Intentionaly left blank



HALAMAN MOTTO

Intentionaly left blank



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan penulisan tesis. Penulis menyampaikan ucapan terima kasih dan penghargaan yang setinggi-tingginya kepada kedua orang tua, mertua, istri dan anak yang telah mendukung dan mendoakan untuk penulis sehingga bisa menyelesaikan kuliah S2 ini. Ucapan terima kasih juga penulis sampaikan kepada:

1. Prof. Dr. M. Suyanto, M.M sebagai rektor Universitas AMIKOM Yogyakarta yang telah memberikan arahan dan dukungan kepada penulis.
2. Dr. Kusrini, M.Kom sebagai direktur Magister Teknik Informatika Universitas AMIKOM Yogykarta
3. Dr. Andi Sunyoto, M.Kom sebagai dosen pembimbing utama dan Drs. Asro Nasiri, M.Kom sebagai dosen pembimbing pendamping yang telah memberikan arahan, bimbingan, inspirasi dan dukungan dalam proses penyelesaian thesis.
4. Seluruh dosen dan staf Magister Teknik Informatika yang telah menjalankan sistem perkuliahan di Universitas Amikom Yogyakarta

Akhirnya dengan segala kekurangan, penulis menyadari bahwa penulisan tesis ini masih butuh banyak perbaikan, sehingga kritik dan saran yang bersifat membangun sangat diharapkan.

Yogyakarta, 1 Agustus 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Ponzi adalah skema penipuan yang sudah ada sejak 150 tahun lalu. Sekarang berkembang cara penawarannya dengan teknologi yang modern. Salah satu teknologi yang digunakan adalah cryptocurrency. Ethereum sebagai blockchain platform yang juga mengakomodasi cryptocurrency menjadi ajang beredarnya penipuan ponzi. Tidak sedikit nilai finansial yang menjadi kerugian pengguna Ethereum Blockchain.

Ponzi yang beredar sangat diuntungkan dengan adanya smart contract. Karena proses transaksi yang bersifat rahasia dan tanpa adanya campur tangan pihak ketiga. Sekali transaksi ponzi dilakukan tidak ada cara untuk mengembalikan ke pengirimnya. Penelitian yang ada mempunyai tingkat akurasi yang bagus tetapi kebanyakan harus berdasarkan data transaksi. Ini artinya deteksi ponzi baru bisa dilakukan jika adanya transaksi.

Dengan kata lain sudah ada korban yang melakukan transaksi. Hal tersebut butuh adanya penelitian deteksi ponzi sebelum jatuh korban. Pada penelitian ini dikemukakan proses yang lebih cepat untuk pendeteksian skema ponzi pada Blockchain Ethereum. Dengan menggunakan XGBClassifier dalam klasifikasi bytecode yang diubah menjadi matrik bobot dari TF-IDF. Ditemukan hasil paling maksimal adalah F1 Score sebesar 98,16%%.

Kata kunci: Blockchain Ethereum, smart contract, bytecode, word embedding, cryptocurrency

ABSTRACT

Ponzi is a fraud scheme since 150 years now. Now different way of offering using modern technology. One of the technologies used is cryptocurrency. Ethereum as a blockchain platform that also accommodates cryptocurrencies, has become a medium for ponzi fraud to circulate. There is not a small amount of financial value lossing for Ethereum users. Ponzi in circulation greatly get benefit from the existence of smart contracts.

Because the transaction process is confidential and without the intervention of a third party. Once a ponzi transaction is made there is no way to return it to the investor. Existing research has a good degree of accuracy but is mostly based on transaction data.

That means ponzi detection can only be done if there is a transaction. In other words, there are victims who make transactions. This requires research to detect ponzi before falling victim. This research proposes a faster process for detecting Ponzi schemes on Ethereum Blockchain. By using TF-IDF to weight smart contract. Then a model is made using the XGBClassifier. The test results for the F1 Score is 98.16%.

Keyword: Ethereum Blockchain, smart contract, bytecode, word embedding, cryptocurrency

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Blockchain adalah teknologi yang menyimpan data berdasarkan urutan catatan-catatan yang saling terkait. Catatan akan disimpan pada satu blok, dimana blok tersebut juga menyimpan hash dari blok sebelumnya. Jika ada perubahan data pada salah satu blok, maka akan mengakibatkan blok berikutnya menjadi tidak lagi valid. Rangkaian blok tersebut akan disebar ke berbagai node atau komputer. Jika terdapat perubahan data pada satu node atau komputer maka akan bisa divalidasi dengan metode konsensus. Hal ini menjadikan teknologi blokchain sangat aman (Namasudra dkk. 2021)

Karena tingkat keamanan blockchain yang tinggi, maka teknologi tersebut digunakan oleh Ethereum Blockchain untuk mengelola cryptocurrency. Cryptocurrency adalah mata uang digital yang terenkripsi dan disimpan dalam rangkaian blockchain serta disebar ke banyak node. Mirip dengan uang konvensional, mata uang digital pada Blockchain Ethereum yang bernama Ether, bisa dijadikan alat jual beli pada lingkungan Blockchain Ethereum (Ferretti dan D'Angelo 2020). Dalam satu hari saja Ether yang digunakan dalam transaksi lebih dari 17 miliar dolar Amerika. Harga satu Ether adalah USD 4.237,82 jika dikonversi ke Dollar Amerika (coinmarketcap.com, 2021). Hal tersebut merupakan bukti bahwa Blockchain Ethereum adalah platform cryptocurrency yang besar.

Salah satu tujuan dibuatnya Blockchain Ethereum adalah agar aplikasi lain bisa dibangun di atas platform Blockchain Ethereum. Selain itu, bertujuan mempermudah transaksi dimana pengguna bisa membuat smart contract sendiri. Smart Contract merupakan kode program yang dibuat sedemikian rupa, sehingga ketika ditemukan kondisi yang sesuai, maka transaksi tertentu akan dieksekusi. Misalnya adalah jika ada pembeli yang membayar kepada penjual dropshipper, maka smart contract akan otomatis mengirim Ether ke akun produsen (Savelyev 2017).

Kelebihan dari smart contract yang tidak perlu pihak ketiga untuk perantara dan verifikasi, banyak digunakan untuk penipuan (Tsankov dkk. 2018). Salahsatunya adalah skema ponzi. Antara tahun 2015 sampai dengan 2017 ditemukan 191 ponzi yang memanfaatkan smart contract pada Blockchain Ethereum. Mengakibatkan kerugian sebesar $500.000 berasal lebih dari 2000 pengguna Ethereum (Bartoletti dkk. 2019).

Secara hukum, kegiatan finansial ponzi adalah penipuan. Biasanya disebutkan sebagai investasi yang akan menghasilkan laba yang sangat besar untuk investor. Pada dasarnya, laba yang diberikan kepada investor adalah dana yang dihimpun dari investor baru. Skema ponzi kebanyakan berbentuk piramida. Investor bawah dari skema piramida akan kehilangan uang dan rugi jika tidak ada investor baru yang bergabung (SEC, 2019).

Penelitian yang dilakukan Liang dkk (2021) menggunakan bytecode dari smart contract dan transaksi sebagai dataset. Dari data yang ada dibuatkan graph yang menghubungkan antar account serta opcode hasil konversi dari bytecode.

Graph yang sudah jadi dianalisa menggunakan Long Short Term Memory sebagai training deteksi ponzi. Hasil yang didapat adalah F-score 91%. Kelemahan dari penelitian ini adalah sebagian dataset berupa transaksi, sehingga contract dianggap ponzi jika sudah ada transaksi. Dengan kata lain pendeteksian terlambat, karena sudah ada korban.

Yu dkk (2021) mengambil dataset transaksi Blockchain ethereum untuk dijadikan data training dan uji. Feature yang telah diambil kemudian dianalisa menggunakan Graph Convolutional Network (GCN) untuk membedakan transaksi ponzi atau bukan. Hasil dari analisa tersebut adalah F-score 89%. Kelemahan dari penelitian ini adalah sebagian dataset berupa transaksi, sehingga contract dianggap ponzi jika sudah ada transaksi. Dengan kata lain pendeteksian terlambat, karena sudah ada korban.

Chen dkk (2021) menggunakan souce code smart contract untuk menjadi obyek penelitian. Source code yang didapat akan diurai dalam bentuk Abstract Syntax Tree (AST). Kemudian bentuk AST dikonversi menjadi urutan Structure-Based Traversal (SBT). Matrix dari bentuk SBT dimasukkan pada convolutional layer. Hasil yang dicapai pada metode ini adalah F-score sebesar 0.89. Kelemahan dari penelitian ini adalah dataset berupa source code. Padahal kebanyakan smart contract bersifat close source. Artinya dataset semakin sedikit dibanding dengan smart contract berupa bytecode.

Shen, Jiang, dan Zhang (2021) menggunakan data bytecode dari smart contract sebagai dataset. Kemudian bytecode tersebut diubah menjadi matrix dimensi banyak. Karena berbentuk dimensi yang besar maka perlu dikurangi

noisenya. Hal ini menggunakan Principal component Analysis untuk menghilangkan noise pada matrix tersebut. Hasilnya dianalisa menggunakan One Class Support Vector Machine (OC-SVM) dan Isolation Forest (Iforest) sebagai metode deteksi anomai. F-Score yang didapat adalah 0.84 untuk OC-SVM dan 0.88 untuk IForest. Melihat score tersebut masih perlunya peningkatan metode.

Penelitian Wang dkk (2021) menggunakan bytecode dan informasi transaksi dari account terkait. Untuk bytecode dikonversi menjadi upcode. Karena dataset bersifat imbalance, maka dilakukan oversampling menggunakan SMOTE. Feature yang digunakan adalah sebanyak 83 buah, yang berasal dari account feature dan code feature. Metode trainingnya menggunakan Long Short Term Memory, dan menghasilkan F-score sebesar 96%. Kelemahan dari penelitian ini adalah sebagian dataset berupa transaksi, sehingga contract dianggap ponzi jika sudah ada transaksi. Dengan kata lain pendeteksian terlambat, karena sudah ada korban.

Fan dkk (2021) menggunakan smart code berupa bytecode sebagai dataset. Konversi bytecode menjadi opcode menggunakan Pyevmasm library. Kemudian opcode dikonversi menjadi unigram. Pada proses pengurutan kata pada unigram ditentukan juga 'kata henti' dari smart contract. Pada unigram dihasilkan 1-gram, 2-gram, 3-gram dan 4-gram kemudian dikonversi menjadi vector berdaskan bag of words. ANOVA F-value digunakan untuk memilih feature yang paling berpengaruh untuk klasifikasi. Untuk mengatasi data imbalance, maka digunakna borderline-SMOTE 2 untuk menghilankan kekurangan tersebut. Hasil pengujian

membuktikan dengan metode ini sebesar 96% untuk F-score. Melihat score tersebut masih perlunya peningkatan metode.

Dari latar belakang di atas bisa dilihat bahwa skema ponzi sangat merugikan banyak orang dengan nilai yang tidak sedikit. Selain itu penelitian dalam pendeteksian skema ponzi dimana hasil yang masih perlu ditingkatkan dan deteksi yang terlambat karena menunggu adanya transaksi, maka perlu adanya penelitian yang lebih bagus dan lebih cepat mendeteksi ponzi.

Penulis mengusulkan metode deteksi skema ponzi pada Blockchain ethereum menggunakan dataset bytecode smart contract, sehingga didapatkan hasil yang lebih bagus dan lebih cepat. Lebih bagus dalam artian mendapatkan akurasi yang lebih tinggi. Lebih cepat dalam artian deteksi ponzi bisa langsung dilakukan hanya dengan melihat bytecode smart contract, sebelum jatuhnya korban dengan melakukan transaksi dengan smart contract tersebut. Word Embedding digunakan untuk menganalisa bytecode yang berupa angka heksadesimal sehingga bisa membedakan apakah smart kontrak tersebut mengandung unsur ponzi atau tidak. Hasil akhir dari metode yang diajukan berupa model.

1.2. Rumusan Masalah

a. Apakah dengan menggunakan smart contract dapat dilakukan klasifikasi skema ponzi?
b. Berapa tingkat akurasi deteksi skema ponzi pada metode TF-IDF dan XGBoost?

c. Parameter n-gram yang optimal untuk mendeteksi skema ponzi?

1.3. Batasan Masalah

a. Deteksi skema ponzi dilakukan pada Blockchain Ethereum.

b. Fitur yang digunakan adalah Smart Contract.

c. Deteksi tidak secara real time.

d. Pengujian menggunakan *Classification Accuracy*.

e. Dataset yang digunakan adalah sama dengan data Chen dkk (2019).

1.4. Tujuan Penelitian

a. Membuktikan bahwa smart contract pada Blockchain ethereum bisa dijadikan dasar klasifikasi skema ponzi atau tidak.

b. Meningkatkan akurasi dari metode yang telah ada.

c. Mencari parameter yang optimal untuk akurasi deteksi.

d. Mendapatkan dataset yang lengkap dalam cakupan alamat smart contract, label ponzi atau bukan, bytecode dan opcode.

1.5. Manfaat Penelitian

a. Ditemukannya metode yang lebih bagus dalam deteksi ponzi pada Blockchain Ethereum.

b. Pencegahan penipuan skema ponzi pada Blockchain ethereum.

c. Membuat model yang bisa mendeteksi skema ponzi berdasarkan smart contract.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian yang telah dilakukan oleh Fan dkk (2021) mendapatkan score F_1 untuk deteksi ponzi pada Ehtereum sebesar 96%. Dataset yang didapat akan dianalisa jika terdapat smart contract yang ganda ataupun yang sudah hancur. Selanjutnya data dilakukan oversampling untuk menghilangkan imbalance.

Pada setiap extraksi feature membutuhkan campur tangan manusia, maka Liang dkk (2021) mengusahakan untuk memininalisirnya. Deteksi akun ponzi menggunakan graph dinamis yang otomatis menggambarkan aktifitas akun ponzi.

Chen, dkk (2021) memilih untuk menggunakan mutli Channel Text CNN untuk agar bisa mengambil struktur dan semantic feature yang hilang jika menggunakan 'hand-crafted feature'.

Penelitian yang dilakukan Yu dkk (2021) menggunakan deep learning untuk deteksi skema ponzi. Algoritma yang dipilih adalah Graph Convolutional network (GCN) diimplementasikan pada smart contract Blockchain Ethereum.

Shen, Jiang, dan Zhang (2021) mengubah bytecode menjadi High-Dimensional matrix. Dari matrix tersusun tersebut dianalisa menggunakan OC-SVM dan IForest dalam pendeteksian skema ponzi pada Ehtereum.

Wang dkk (2021) menggabungkan feature dari account dan smart contract untuk dianalisa menggunakan oversampling based Long Short Term Memory

(LSTM). Featrue dari account yang diambil adalah jumlah saldo yang dimiliki akun terkait.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian

Deteksi Skema Ponzi pada Blochain Menggunakan Algoritma Word Embedding

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	AI-SPSD: Anti-Leakage Smart Ponzi Schemes Detection in Blockchain	Fan, Shuhui, Shaojing Fu, Haoran Xu, dan Xiaochun Cheng. Information Processing & Management. 2021.	Mendeteksi skema ponzi dari data yang imbalance, ganda, dan pergeseran akurasi akibat perhitungan gradien.	Menghasilkan score F_1 sebesar 96%.	Diperlukannya pengembangan deteksi secara realtime, serta skema penipuan selain ponzi.	Hasil yang diharapkan dari penelitian yang akan dilakukan minimal sama atau bahkan lebih baik dengan perhitungan feature lebih sedikit.
2	Data-Driven Smart Ponzi Scheme Detection	Liang, Yuzhi, Weijing Wu, Kai Lei, dan Feiyang Wang. Arxiv. 2021.	Mendeteksi skema ponzi dengan mengurangi campur tangan manusia	Metode yang dibangun bisa mendeteksi akun ponzi menggunkan graph dinamis yang otomatis menggambarkan aktifitas akun target.	Pembobotan network sebatas jmlah transaksi belum pada 'value' tiap transaksi.	Tidak menggunakan deep learning, sehingga model tidak perlu pengaturan environment.

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Improving Ponzi Scheme Contract Detection Using Multi-Channel TextCNN and Transformer	Chen, Yizhou, Heng Dai, Xiao Yu, Wenhua Hu, Zhiwen Xie, dan Cheng Tan. Sensors. 2021.	Mendeteksi skema ponzi berdasarkan smart contract.	Berhasil mendeteksi skema ponzi tanpa menggunakan 'Hand-crafted feature' sehingga bisa mengambil struktur dan feature semantik.	Perlu diimplementasikannya algoritma klasifikasi yang lain.	Tidak perlu metode extract feature yang otomatis sehingga mengurangi akurasi dari feature itu sendiri.
4	Ponzi Scheme Detection in EthereumTransaction Network	Shen, dan Qi Xuan. ArXiv. 2021.	Mendeteksi skema ponzi pada Ethereum	Menggunakan Graph Convolutional network (GCN) untuk membedakan kontrak ponzi dalam prespektif network.	Belum bisa mendeteksi ponzi skema secara real time.	Tidak menggunakan deep learning, sehingga model tidak perlu pengaturan environment.
5	Mining Bytecode Features of Smart Contracts to Detect Ponzi Scheme on Blockchain	Shen, Xiajiong, Shuaimin Jiang, dan Lei Zhang. Computer Modeling in Engineering & Sciences. 2021	Mendeteksi skema ponzi pada smart contract Ethereum.	Dataset yang berupa bytecode diubah menjadi high-dimensional matrix. Menggunakan algoritma OC-SVM dan IFOREST.	Perlu deteksi yang realtime dan menambah feature agar lebih akurat.	Hasil yang diharapkan dari penelitian yang akan dilakukan minimal sama atau bahkan lebih baik dengan perhitungan feature lebih sedikit.

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Ponzi Scheme Detection via Oversampling-Based Long Short-Term Memory for Smart Contracts	Wang, Lei, Hao Cheng, Zibin Zheng, Aijun Yang, dan Xiaohu Zhu. Knowledge-Based Systems. 2021.	Mendeteksi skema ponzi pada smart contract Ethereum.	Menggabungkan account feature dan contract code kemudian diolah menggunakan Oversampling Based Long Short Term Memory (LSTM).	Perlunya penggunaan Deep Learning dalam proses pembuatan model.	Hasil yang diharapkan dari penelitian yang akan dilakukan minimal sama atau bahkan lebih baik dengan perhitungan feature lebih sedikit.

2.3. Landasan Teori

2.3.1. Blockchain

Blockchain adalah teknologi peer-to-peer, catatan terdistribusi yang diamankan dengan kriptografi, pencatatan tanpa bisa dihapus, (hampir) tidak bisa di-hack, dan hanya bisa diubah melalui konsensos atau kesepakatan antar peers (Bashir 2020).

Pada tahun 1991 Haber dan Stornetta pertama kali mengusulkan rangkaian kriptografi yang aman. Di sini, komputer mengirim dokumen ke sebuah server waktu untuk diberi stempel waktu dan server menandatangani dokumen dan stempel waktu tersebut dalam bentuk hash. Data tersebut dijadikan rujukan oleh pointer, tetapi bukan berupa alamat dokumen. Jadi, ketika data diubah, pointer menjadi tidak valid, dan tidak ada yang bisa mengubah data di server. Konsep ini hampir mirip dengan konsep Blockchain.

Finney telah memperkenalkan konsep "*reusable proofs of work*" untuk memvalidasi data yang berupa rangkaian (chain). Namun, sistem Finney masih berupa sistem terpusat belum terdistribusi.

Nakamoto pertama kali memperkenalkan ide Blockchain yang diimplementasikan untuk cryptocurrency. Bitcoin adalah mata uang digital pertama, dan konsep Bitcoin menginspirasi banyak aplikasi. Pada Era tersebut merupakan generasi pertama dari Blockchain yang dikenal dengan Blockchain 1.0. Kemudian muncul Blockchain 2.0 yang dapat disebut sebagai aplikasi baru Blockchain, dan pertama kali muncul pada tahun 2014. Di Blockchain 2.0, para

programmer dapat menulis smart contract mereka sendiri dan mengembangkan aplikasi mereka sendiri.

Pada generasi ini, dimungkinkan untuk menyimpan ID digital pengguna dan bisa menyelesaiian permasalah pada berbagai transaksi penting. Pada tahun 2014, ukuran file Blockchain Bitcoin mencapai 20 GB, yang berisi informasi semua transaksi. Pada tahun 2015, ukurannya hampir 30GB, dan pada tahun 2017, ukurannya mencapai 100GB. IBM membuka pusat penelitian Blockchain di Singapura pada Juli 2016. Di era Blockchain 3.0, para peneliti berusaha meminimalkan waktu proses transaksi.

Gambar 2.1. Struktur dasar blockchain

Beberapa kelebihan blockchain sehingga menjadikannya popupler

a. Keamanan: Keamanan adalah salah satu isu utama dunia digital saat ini. Di Blockchain, semua transaksi yang divalidasi atau dieksekusi disimpan secara permanen di blok yang tidak dapat dihapus atau diubah oleh siapa pun.
b. Terdistribusi: Teknologi Blockchain mendukung banyak komputer yang didistribusikan dari seluruh dunia. Misalnya, Blockchain Ethereum

mendukung hampir semua pengguna atau individu untuk menyumbangkan jaringan mereka hanya dengan menginstal perangkat lunak mereka.

c. Efisiensi: Efisiensi jaringan dapat ditingkatkan dengan menggunakan teknologi Blockchain, ketika insttitusi-institusi keuangan berkolaborasi di antara mereka sendiri. Dengan menggunakan teknologi Blockchain, sebuah sistem dapat dikembangkan untuk meningkatkan efisiensi jaringan secara real-time.
d. Transparansi: Dalam jaringan Blockchain, data dipublikasikan pada platform umum, dan pihak yang berkepentingan serta regulator dapat dengan mudah mendapatkan tampilan real-time dari platform tersebut.
e. Ketahanan: Dalam teknologi Blockchain, bahkan dengan jumlah peserta atau individu yang sangat besar, ketangguhan data meningkat dengan umur yang lebih panjang.
f. Kepercayaan: Mayoritas individu atau peserta harus menyetujui data sebelum menambahkannya ke jaringan Blockchain, yang berbeda dari jaringan yang hanya dipusatkan pada satu titik atau satu pihak saja. Dengan demikian, kepercayaan meningkat untuk membuat, mengubah, atau bahkan membaca data apa pun.

Disamping berbagai kelebihan dari blockcahin terdapat juga beberapa kelemahan atau kekurangan, yaitu:

a. Pemborosan: Dalam teknologi Blockchain, setiap node harus menjalankan atau memelihara algoritma konsensus, yang memberikan kemampuan nol

toleransi kesalahan. Namun, semua ini sia-sia karena setiap node mengikuti tugas yang sama untuk mencapai konsensus.

b. Kecepatan dan Biaya Jaringan: ketika jumlah node bertambah banyak, maka semkin sulit untuk mengelola node tersebut. Di sini, transaksi menjadi lambat karena sebagian node mendapatkan prioritas dalam hal imbalan, dan catatan transaksi semakin menumpuk.
c. Ukuran Blok: Di Blockchain, setiap blok atau transaksi ditambahkan ke rantai atau jaringan yang meningkatkan ukuran basis data.
d. Kinerja: Jaringan Blockchain selalu lebih lambat dari *cetralized database*. Ketika transaksi dijalankan, blockchain menjalankan semua proses database reguler bersama dengan banyak beban tambahan seperti validasi signature, algoritma konsensus, dll.
e. Standar: Karena Blockchain masih dalam usia dini, tidak ada standar khusus. Standar yang bagus dapat meningkatkan keamanan jaringan Blockchain.

Secara umum terdapat tiga jenis blockchain, yaitu:

a. *Public Blockchain*: Seperti namanya, *Public Blockchain* adalah Blockchain yang ditangani banyak pihak. Tidak ada seorang pun yang bertanggung jawab atau mengontrol, dan setiap orang dapat berpartisipasi dalam jaringan untuk mengubah, membaca, atau mengaudit data. Public Blockchain sifatnya terbuka dan transparan untuk semua orang. Karena tidak ada administrator pusat atau penanggung jawab, semua keputusan diambil sesuai dengan berbagai algoritme konsensus terdesentralisasi, seperti Proof of Stake (PoS), Proof of Work (PoW), dan banyak lagi. Contoh: Litecoin dan Bitcoin

b. *Private Blockchain*: adalah aset pribadi dari suatu organisasi atau individu. Berbeda dengan *Public Blockchain*, ada biaya yang besar dalam penangan jaringan blockchain. Di sini, algoritma konsensus dikembangkan dengan penanggung jawab pusat, yang dapat memberikan hak penambangan kepada siapa pun. *Private Blockchain* diamankan menggunakan kriptografi dan sangat efektif.

c. *Consortium* atau *Federated Blockchain*: Sebuah konsorsium Blockchain mencoba menghilangkan kelemahan *Private Blockchain*, di mana satu entitas bisa mendapatkan semua hak. Ada lebih banyak entitas yang bertanggung jawab daripada satu yang bertanggung jawab pada blockchain jenis ini. Di sini, sekelompok perwakilan atau individu atau perusahaan berkumpul dan mengambil semua keputusan untuk manfaat keseluruhan dari jaringan Blockchain. Konsorsium Blockchain kinerja nya cepat tetapi terdapat banyak titik potensi kesalahan atau celah keamanan.

2.3.2. Cryptocurrency

Salah satu penemuan teknologi terbesar di dunia modern yang menarik banyak perhatian publik adalah fenomena cryptocurrency. Menurut beberapa pendapat, ini adalah penemuan teknologi terbesar dalam sepuluh tahun terakhir. Cryptocurrency menjadi sangat populer dalam waktu yang sangat singkat. Cryptocurrency mewakili aset digital, yang tujuan utamanya adalah menjadi media pertukaran atau transaksi. Pada saat melakukan pertukaran atau transaksi

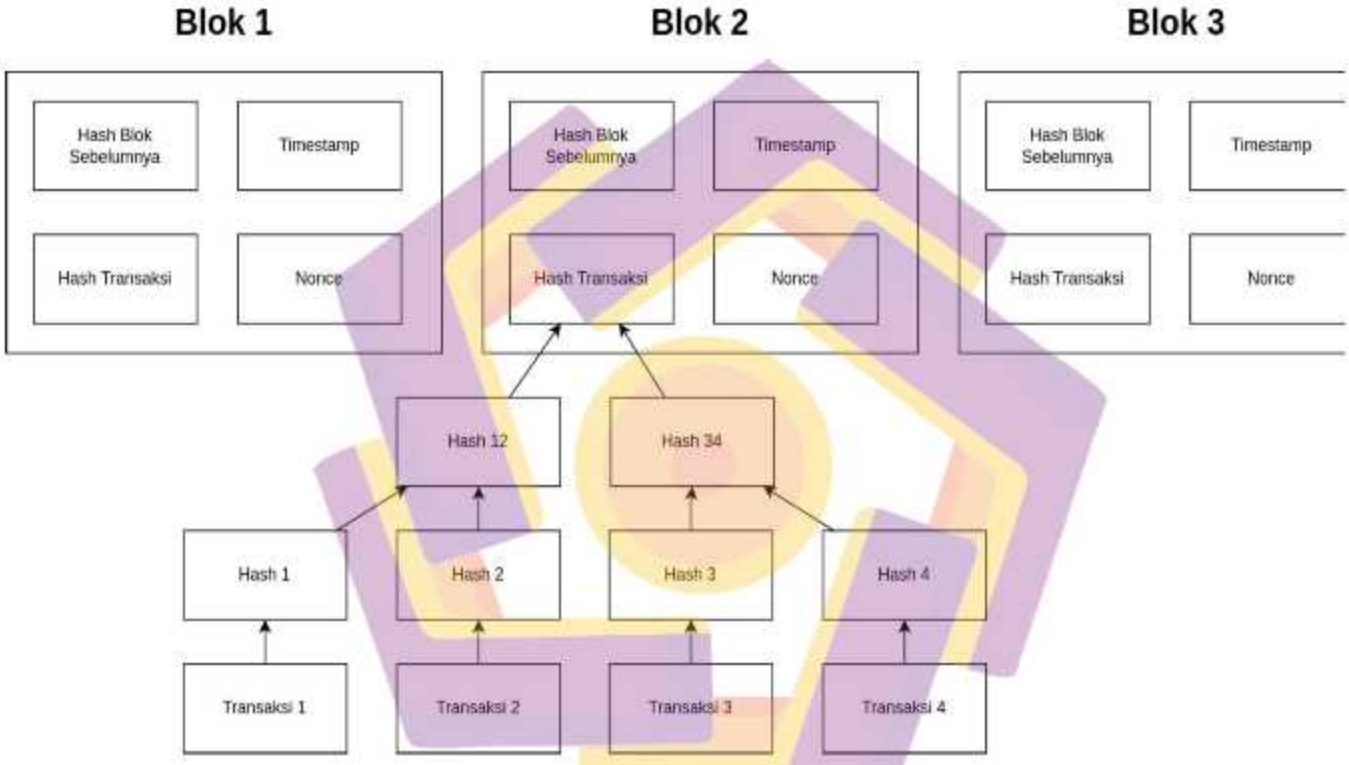
itu, cryptocurrency menggunakan kriptografi sehingga semua transaksi diamankan, semua yang baru buat selalu dikendalikan oleh sistemnya sendiri.

Dapat dikatakan bahwa cryptocurrency adalah bagian dari mata uang digital. Cryptocurrency pertama yang pernah dibuat adalah Bitcoin, pada tahun 2009. Setelah itu, banyak cryptocurrency lain muncul di pasar, tetapi mereka disebut altcoin, karena mereka mewakili campuran alternatif Bitcoin.

Bitcoin tidak memiliki sistem terpusat, tidak ada yang bisa mengendalikannya sepenuhnya, tidak seperti pada sistem perbankan elektronik. Dalam sistem perbankan, kami memiliki lembaga yang dapat mengeluarkan mata uang dan mencetak uang. Tetapi hal-hal berbeda dengan cryptocurrency. Mereka menggunakan kriptografi untuk mengumpulkan semua informasi dan data, dan semuanya melewati blockchain, yang mewakili buku besar yang didistribusikan.

Pemerintah tidak memiliki kekuatan untuk memproduksi unit baru, semuanya dikendalikan melalui buku besar digital virtual. Masih belum diketahui siapa yang membuat Bitcoin, satu-satunya hal yang diketahui publik adalah bahwa seseorang atau mungkin sekelompok orang dengan nama Satoshi Nakamoto yang mewujudkannya.

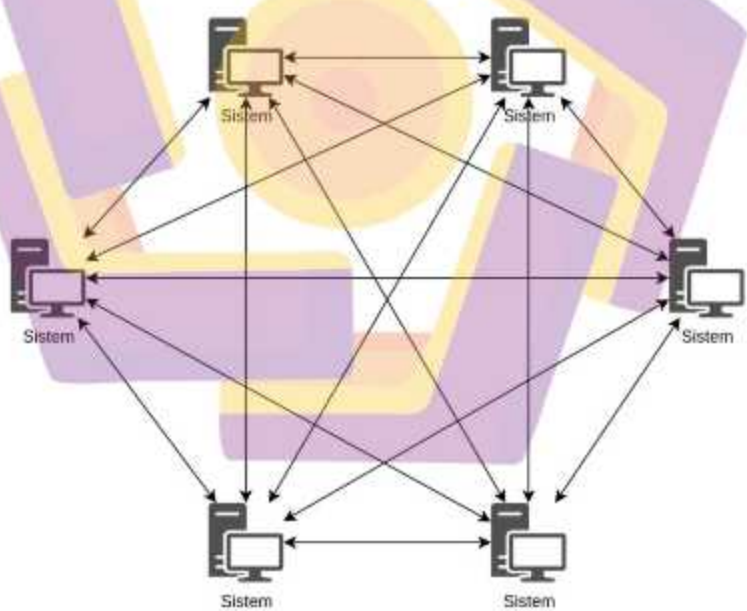
Agar sistem ini berfungsi, ada banyak orang dari masyarakat umum yang disebut penambang (Nakamoto, 2008). Tugas mereka adalah menggunakan komputer mereka untuk validasi dan transaksi stempel waktu, karena mereka menambahkannya ke buku besar saat menggunakan skema signature berdasarkan waktu timestamp. Dan juga mereka mendapatkan insentif besar untuk pekerjaan ini.



Gambar 2.2. Konsep dari Markle Tree

2.3.3. Ehtereum

Ethereum adalah blockhain platform yang berbasis opensource, sistem yang ter-desentralisasi dengan fitur unggulan smart contract. Termasuk cryptocurrency dengan nama mata uang Ether. Dibuat oleh Vitalik Buterin pada tahun 2013, dan berjalan online pada 2015. Berbeda dengan Bitcoin yang fokus pada pembayaran cryptocurrency, Ethereum bisa dijadikan platform dasar untuk membangung aplikasi desentralisasi yang lain. Misalnya adalah pembelian tiket konser atau marketplace, sehingga Ethereum diintegrasikan dengan aplikasi custom yang dibuat oleh pengembang lain (Grincalaitis 2019).



Gambar 2.3. Jaringan Peer to Peer

Ethereum adalah platform blockchain yang terkenal. Tidak seperti Bitcoin, ia menyediakan Mesin Virtual Ethereum (EVM) yang dapat mengeksekusi secara bebas kode algoritma yang komplek. EVM adalah wadah runtime untuk bytecode EVM, yang merupakan bahasa pemrograman berkonsep complete-Turing. Untuk membantu membangun aplikasi, banyak bahasa tingkat tinggi baru seperti Solidity dibuat. Hanya dibutuhkan beberapa baris kode untuk membuat aplikasi terdistribusi sederhana dengan platform Ethereum.

Smart Contract merupakan seubah ide yang dicetuskan oleh Nick Szabo pada tahun 1996, bertujuan untuk menyediakan cara baru untuk melegalkan transaksi bisnis atau hubungan antar individu. Sangat gampang penerapan smart contract pada platform Ethereum dikarenakan adanya EVM tersebut.

Secara teknis, hanya dibutuhkan tiga langkah untuk mengimplementasikan smart contract pada Ethereum:

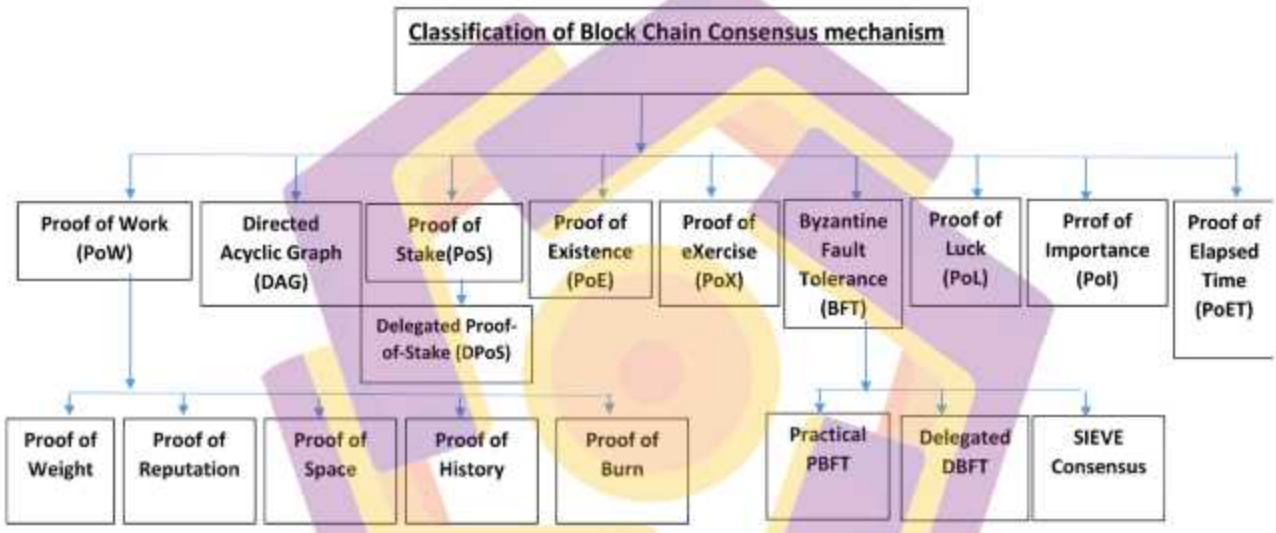
a. Menulis source code smart contract dengan bahasa tingkat tinggi, seperti Solidty.
b. Mengkompilasi source code menjadi bytecode menggunakan kompiler EVM.
c. Mengunggah bytecode ke blockchain Ethereum dengan klien Ethereum.

Dari sudut pandang teknis, sistem blockchain dapat dianggap sebagai sistem yang memindahkan status serta memeliharanya. Di blockchain Ethereum, status bisa mencakup data akun beserta status akun tersebut. Akun, atau alamat alamat akuan, adalah rangkaian angka dan karakter yang dapat dibagikan dengan siapa saja yang ingin bertransaksi dengan akun tersebut.

Dua jenis akun pada Ethereum: akun eksternal yang dikendalikan oleh pengguna (yaitu, manusia) dan akun kontrak yang dikendalikan oleh bytecode yang disimpan bersama dengan akun. Terlepas dari jenis akun, mereka diperlakukan sama oleh EVM. Setiap akun memiliki dua data yang menentukan status unik akun: nonce (menunjukkan jumlah transaksi yang dikirim dari alamat tersebut) dan saldo.

Transaksi biasanya merupakan pengiriman pesan dari satu akun ke akun lain dengan data biner atau Ether (gas kripto untuk jaringan Ethereum). Mentransfer Ether dan memanggil fungsi dalam kontrak adalah dua transaksi dasar yang dimulai oleh pengguna. Selain itu, transaksi baru dapat dipicu oleh transaksi yang sudah ada. Misalnya, korban yang menginvestasikan sebagian Ether ke akun kontrak skema Ponzi dapat memicu transaksi transfer Ether dari akun kontrak ke korban sebelumnya.

Transaksi pengiriman Ether ke akun kontrak disebut sebagai transaksi eksternal (atau investasi) dan transaksi transfer yang dipicu sebagai transaksi internal (atau pembayaran) kontrak. Transaksi eksternal dapat dilihat sebagai pemicu pengoperasian smart contract. Hal tersebutt Biasanya dipicu oleh pengguna (yaitu, akun eksternal) dengan beberapa data biner untuk memanggil fungsi dalam kontrak. Transaksi internal adalah semacam reaksi terhadap transaksi eksternal.



Gambar 2.4. Jenis-jenis konsesus pada blockchain

Misalnya, smart contract B akan mentransfer beberapa Ether ke akun C dan D ketika smart contract menerima beberapa Ether dari akun lain. Kemudian, jika akun A mengirim 5 Ether ke akun B, maka akan memicu dua transaksi transfer dari B ke C dan D. Dalam contoh ini, transaksi dari A ke B adalah transaksi eksternal kontrak B; dan transaksi dari B ke C dan D adalah dua transaksi internal kontrak B. Dengan demikian, transaksi eksternal dan internal dapat dilihat sebagai reaksi panggilan untuk smart contract, yang mungkin berguna dalam mencerminkan logika kontrak.

Setiap transaksi dibebankan dengan sejumlah gas, sesuai dengan sumber daya yang dikonsumsi yang digunakan untuk memvalidasi transaksi. Biaya gas tersebut untuk mendorong penambang mau memvalidasis transaksi sehingga mempertahankan blockchain tetap berjalan dan untuk mencegah penyalahgunaan sistem.

Gas dikalikan dengan Gasprice yang ditentukan pengguna adalah biaya untuk transaksi yang harus dibayar di muka dari saldo pengirim dan habis secara bertahap sesuai dengan aturan tertentu. Gas sisa akan dikembalikan dengan cara yang sama setelah eksekusi. Jika gas habis sebelum eksekusi selesai, transaksi gagal dan semua hasil dikembalikan seperti semua.

Oleh karena itu, sangat penting bagi pembuat smart contract untuk menyediakan gas yang cukup untuk memastikan keberhasilan suatu transaksi. Karena setiap transaksi dieksekusi dan dikemas secara independen oleh penambang, dan protokol konsensus harus digunakan untuk mempertahankan status unik blockchain Ethereum dalam periode waktu tertentu.

Protokol konsensus adalah dasar dari rasa saling percaya antara pengguna dalam sistem blockchain. Ethereum mengadopsi protokol konsensus yang serupa dengan Bitcoin. Ini memastikan bahwa sistem Ethereum akan aman kecuali penyerang tertentu memiliki kekuatan komputasi 51% dari seluruh jaringan. Berdasarkan keamanan sistem, kepercayaan dibangun di antara pengguna Ethereum untuk proses eksekusi kode yang transparan.

Tabel 2.2. Contoh source code yang mengandung skema ponzi, nama program ponzinya adalah Rubixi

Source Code

```
pragma solidity ^0.4.15;

contract Rubixi {
//Declare variables for storage critical to contract
uint private balance = 0;
uint private collectedFees = 0;
uint private feePercent = 10;
uint private pyramidMultiplier = 300;
uint private payoutOrder = 0;
address private creator;

//Sets creator
function DynamicPyramid() {
    creator = msg.sender;
}

modifier onlyowner {
    if (msg.sender == creator) _;
}

struct Participant {
    address etherAddress;
    uint payout;
}

Participant[] private participants;

//Fallback function
function() {
    init();
```

```
}

//init function run on fallback
function init() private {
      //Ensures only tx with value of 1 ether or greater are
processed and added to pyramid
    if (msg.value < 1 ether) {
        collectedFees += msg.value;
        return;
    }
    uint _fee = feePercent;
    //50% fee rebate on any ether value of 50 or greater
    if (msg.value >= 50 ether) _fee /= 2;
    addPayout(_fee);
}

//Function called for valid tx to the contract
function addPayout(uint _fee) private {
    //Adds new address to participant array
        participants.push(Participant(msg.sender,  (msg.value  *
pyramidMultiplier) / 100));

     //These statements ensure a quicker payout system to later
pyramid entrants, so the pyramid has a longer lifespan
    if (participants.length == 10) pyramidMultiplier = 200;
    else if (participants.length == 25) pyramidMultiplier = 150;

    // collect fees and update contract balance
    balance += (msg.value * (100 - _fee)) / 100;
    collectedFees += (msg.value * _fee) / 100;

    //Pays earlier participiants if balance sufficient
    while (balance > participants[payoutOrder].payout) {
        uint payoutToSend = participants[payoutOrder].payout;
        participants[payoutOrder].etherAddress.send(payoutToSend
);

        balance -= participants[payoutOrder].payout;
        payoutOrder += 1;
    }
}

//Fee functions for creator
function collectAllFees() onlyowner {
    if (collectedFees == 0) throw;
    creator.send(collectedFees);
    collectedFees = 0;
}
```

```
function collectFeesInEther(uint _amt) onlyowner {
    _amt *= 1 ether;
    if (_amt > collectedFees) collectAllFees();
    if (collectedFees == 0) throw;
    creator.send(_amt);
    collectedFees -= _amt;
}

function collectPercentOfFees(uint _pcent) onlyowner {
    if (collectedFees == 0 || _pcent > 100) throw;
    uint feesToCollect = collectedFees / 100 * _pcent;
    creator.send(feesToCollect);
    collectedFees -= feesToCollect;
}

//Functions for changing variables related to the contract
function changeOwner(address _owner) onlyowner {
    creator = _owner;
}

function changeMultiplier(uint _mult) onlyowner {
    if (_mult > 300 || _mult < 120) throw;
    pyramidMultiplier = _mult;
}

function changeFeePercentage(uint _fee) onlyowner {
    if (_fee > 10) throw;
    feePercent = _fee;
}

//Functions to provide information to end-user using JSON
interface or other interfaces
function currentMultiplier() constant returns(uint multiplier,
string info) {
    multiplier = pyramidMultiplier;
        info = 'This multiplier applies to you as soon as
transaction is received, may be lowered to hasten payouts or
increased if payouts are fast enough. Due to no float or
decimals, multiplier is x100 for a fractional multiplier e.g.
250 is actually a 2.5x multiplier. Capped at 3x max and 1.2x
min.';
}

function currentFeePercentage() constant returns(uint fee,
string info) {
    fee = feePercent;
     info = 'Shown in % form. Fee is halved(50%) for amounts
equal or greater than 50 ethers. (Fee may change, but is capped
to a maximum of 10%)';
```

```
}

function      currentPyramidBalanceApproximately()      constant
returns(uint pyramidBalance, string info) {
    pyramidBalance = balance / 1 ether;
    info = 'All balance values are measured in Ethers, note that
due to no decimal placing, these values show up as integers
only, within the contract itself you will get the exact decimal
value you are supposed to';
}

function       nextPayoutWhenPyramidBalanceTotalsApproximately()
constant returns(uint balancePayout) {
    balancePayout = participants[payoutOrder].payout / 1 ether;
}

function     feesSeperateFromBalanceApproximately()     constant
returns(uint fees) {
    fees = collectedFees / 1 ether;
}

function totalParticipants() constant returns(uint count) {
    count = participants.length;
}

function     numberOfParticipantsWaitingForPayout()     constant
returns(uint count) {
    count = participants.length - payoutOrder;
}

function   participantDetails(uint   orderInPyramid)   constant
returns(address Address, uint Payout) {
    if (orderInPyramid <= participants.length) {
        Address = participants[orderInPyramid].etherAddress;
        Payout = participants[orderInPyramid].payout / 1 ether;
    }
}
}
```

2.3.4. Smart Contract

Smart contract adalah porgram yang bisa mengeksekusi dirinya sendiri ketika ada kondisi pemicu yang telah diatur. Disimpan secara publik pada semua node ethereum sehingga tidak bisa diubah setelah diupload. Sebagai contohnya

adalah ketika sebuah fungsi yang mengirim pesan kepada akun tertentu akan mengirimkan Ether pada akun lain jika kondisi logikanya terpenuhi (Atzei, Bartoletti, dan Cimoli 2017). Isi dari smart contract berupa angka bytecode dimana setiap dua digit merupakan sebuah instruksi dan data. Format ini bisa diubah menjadi berformat opcode seperti pada gambar 2.5.

```
6060604052600080        1   PUSH1 0x60
8055600181905560        2   PUSH1 0x40
0a60025561012c60        3   MSTORE
0355600481905561        4   CALLDATASIZE
09db90819061002d        5   ISZERO
90396000f3606060        6   PUSH2 0x00b9
```

Gambar 2.5. Contoh smart contract berupa bytecode (kiri) dan dikonversi menjadi opcode (kanan). (Chen dkk 2019)

Perbedaan smart contract yang mengandung skema ponzi dengan yang tidak mengandung skema ponzi adalah adanya urutan perintah dan data dengan pola tertentu sehingga akan terjadi rangkaian aliran ether dari satu akun ke akun yang lain dimana akun terakhir akan lebih sedikit mendapatkan kiriman ether dibanding yang lain. Kemunculan perintah dan data tertentu akan mempunyai bobot jika disandingkan dengan perintah dan data selanjutnya. Sehingga bisa ditarik keseimpulan bahwa urutan perintah dan data bisa digunakan untuk membedakan skema ponzi atau tidak pada smart contract (Chen dkk 2019).

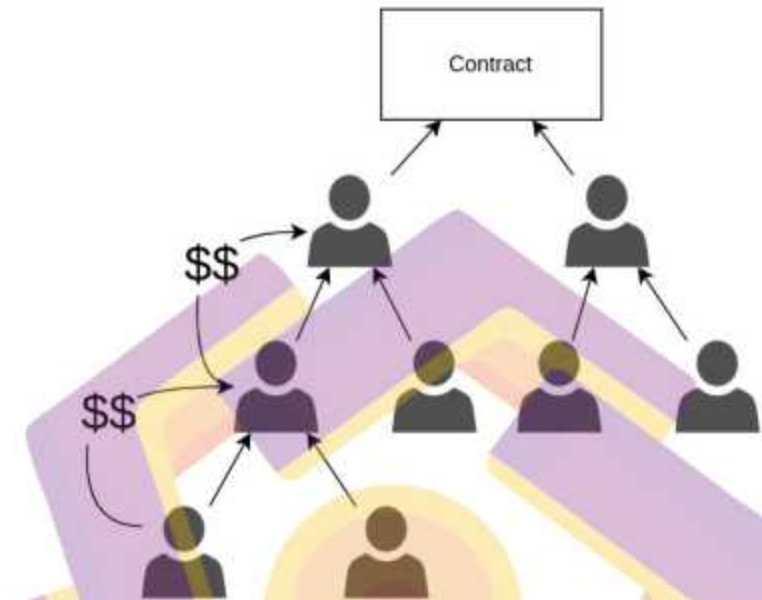
Skema ponzi pada blockchain tidak berbeda dengan skema ponzi konvensional (non-cryptocurrency). Konsep utamanya adalah investor yang telah mengirimkan ether ke smart contract akan mendapatkan keuntungan jika ada investor baru yang masuk. Pola alur ponzi terdapat empat bentuk yaitu piramida berbasis array, piramida berbasis pohon, handover dan air terjun. Kerugian investor adalah jika tidak ada lagi investor baru yang masuk (Bartoletti dkk. 2019).

2.3.5. Word Embedding

Word embedding adalah metode yang mengubah urutan kata dalam kalimat menjadi vektor berdasarkan kedekatan dengan kata yang terdekat. Hal ini akan bisa membedakan suatu kata yang disambung dengan kata lain secara semantik. Vektor yang tersusu berupa matrik yang setiap elemnnya mempunyai bobot. Pada hasil vektornya akan bisa diketahui kata yang paling banyak korelasinya (Jurafsky, 2007).

2.3.6. Ponzi

Skema ponzi adalah skema insvestatsi yang menggunakan dana investasi dari anggota baru untuk membayar laba kepada investor-investor sebelumnya. Investor paling atas atau pembuat investasi pertama akan mendapatkan kompensasi atas bergabungnya anggota baru. Pada anggota level 1 mendapatkan komepenasasi pada anggota level 2 dan seterusnya samapi pada tahap tidak ada anggota baru yang bergabung. Sehingga skema tersebut macet.



Gambar 2.6. Konsep dasar skema ponzi

2.3.7. TF-IDF

TF-IDF atau term frequency–inverse document frequency, metode pembobotan kata pada dokumen sehingga bisa diketahui kata yang penting atau tidak berdasarkan berapa kali kemunculuna kata tersebut dan berada di dokumen mana. Adapun rumus dari TF-IDF adalah:

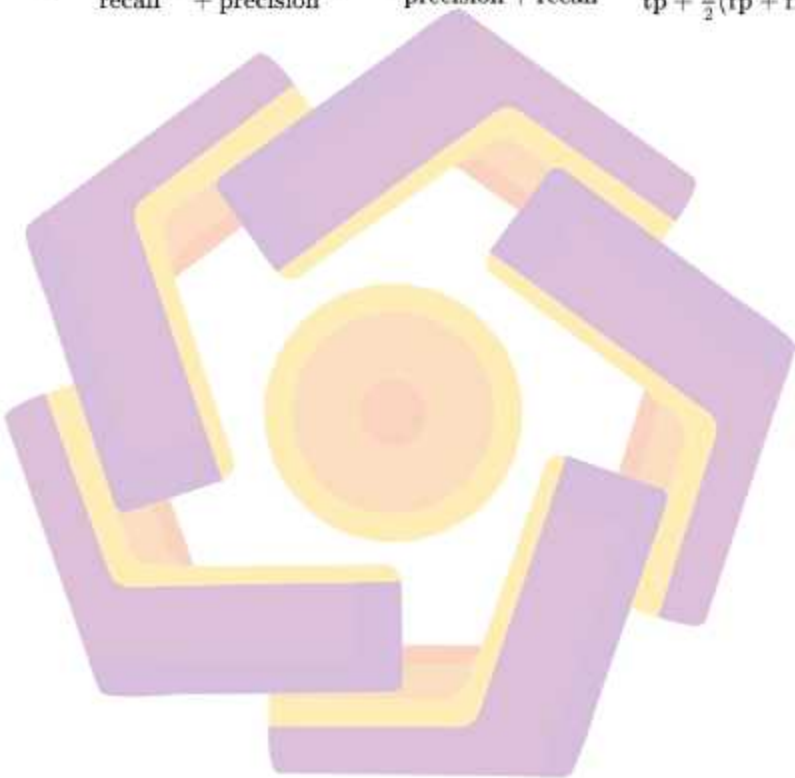
$$tf\text{-}idf(t,d) = tf(t,d) \times idf(t) \quad \text{.....(1)}$$

2.3.8. Pengukuran F-Score

F-Score adalah pengukuran yang memadukan antara *precission* dan *recall*.

Adapun rumusnya adalah sebagai berikut:

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}$$



BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis, sifat dan pendekatan pada penelitian yang akan dilakukan adalah sebagai berikut:

a) Jenis dan Pendekatan Penelitian

Jenis dan pendekatan penelitian adalah penelitian kuantitatif.

b) Sifat Penelitian

Sifat penelitian adalah penelitian eksperimental terapan. Penelitian ini menemukan metode deteksi skema ponzi pada Ethereum.

3.2. Metode Pengumpulan Data

Data yang digunakan adalah dataset yang sudah digunakan oleh penelitian sebelumnya Chen dkk (2019). Pada penelitian tersebut dilakukan pemeriksaan source code secara manual dari smart contract sehingga bisa ditentukan mengandung ponzi atau tidak. Dikarenakan yang didapat dari penelitian sebelumnya hanya berupa alamat smart contract, maka perlunya mendapatkan bytecode dan opcode di tempat lain. Untuk bytecode bisa diambil dari bigquery google, sedangkan opcode diambil dari etherscan.io.

3.3. Metode Analisis Data

Dataset yang telah didapat akan dianalisa dan dibersihkan dari data yang ganda dan tidak lengkap. Pada dasarnya bytecode smart contract ethereum tersusun dari kode hexadecimal yang terdiri per dua karakter. Setiap dua karakter mempunyai arti tersendiri dalam proses eksekusi. Sehingga data bytecode perlu dipecah per dua karakter sehingga proses tokenisasi bisa dilakukan.

Adapun data berupa opcode dibersihkan dari syntax yang tidak perlu. Karena hasil dari etherscan.io tercampur dengan syntax html, maka perlu dihilangkan. Selain itu, smart contract dalam bentuk opcode mengandung alamat akun, hash transaksi dan beberapa kode heksa desimal yang merujuk alamat memory RAM.

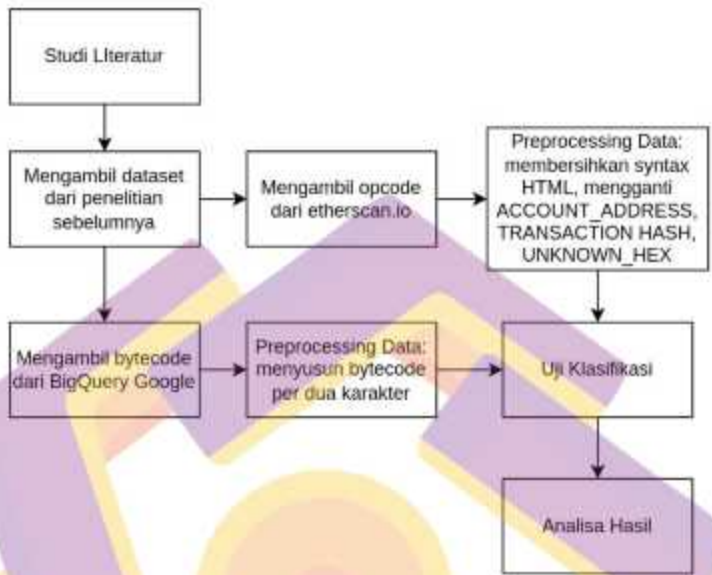
Dari kondisi tersebut alamat akun perlu diganti dengan kata sebagai pengganti. Pada penelitian ini diganti dengan dengan kata ACCOUNT ADDRESS. Sedangkan kode hash transaksi diganti dengan TRASACTION_HASH, dan kode heksadesimal yang lain diganti dengan UNKNOWN_HEX.

3.4. Alur Penelitian

Alur penelitian yang dilakukan adalah sebagai berikut dan digambarkan pada gambar 3.7:

1. Melakukan studi literatur dengan cara mencari naskah penelitian-penelitian sebelumnya serta berbagai rujukan ilmiah terkait blockchain, ethereum, smart contract dan ponzi.

2. Mengambil alamat smart contract dari penelitian sebelumnya
3. Mendownload data bytecode dari bigquery berdasarkan alamat smart contract penelitian sebelumnya. Serta mendownload opcode dari etherscan.io
4. Membersihkan data opcode dari syntax html
5. Menghapus data alamat smart contract opcode ataupun bytecode yang null.
6. Mengubah alamat akun dalam opcode menjadi kata ACCOUNT_ADDRESS, dan hash transaksi menjadi TRASACTION_HASH.
7. Mengubah kode heksadesimal pada opcode selain alamat akun dan hash transaksi menjadi UNKNOWN_HEX
8. Melakukan uji coba klasifikasi menggunakan berbagai metode beserta mebandingkan data yang imbalanced dan sudah balanced.
9. Melakukan analisa hasil dari uji coba tersebut.



Gambar 3.7. Alur Penelitian

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pengambilan Dataset

Dalam penelitian ini digunakan dataset yang berasal dari penelitian Chen dkk (2019), file yang didapat adalah berformat CSV berisi kolom alamat smart contract dan label seperti pada tabel 4.3. Alamat smart contract berupa karakter 42 digit heksadesimal. Sedangkan label berupa angka 0 untuk smart contrat bukan ponzi dan angka 1 untuk smart contract berisi skema ponzi. Perbandingan data yang berskema ponzi dan bukan adalah 200:3594, sehingga data ini tidak imbang.

Klasifikasi ponzi hanya terdiri dari dua kelas atau label, yaitu kelas atau label 1 menunjukkan bahwa smart contract tersebut mengandung ponzi. Sedangkan kelas atau label 0 menunjukkan bahwa alamat smart contract tersebut mempunyai bytecode atau opcode yang mengandung skema ponzi.

Tabel 4.3. Contoh dataset yang didapat dari Chen dkk (2019)

Contract	Ponzi
0xf41624c6465e57a0dca498ef0b62f07cbaab09ca	0
0xf77cead5b85f379aa7b9031a842ccd903d7966a7	0
0x582b2489710a4189ad558b6958641789587fcc27	1
0x5745b21F6365b379451F2f327274812D7A5D3d0D	0
0xB71Cd8f21D5C21edcd113CA2807189ddcAf9DD44	1
0xEb4245C88C660Ae4eE23c76054e5490ccd7bbd82	0
0x8e911d478da4f4ddc3a6dd76b3fcdd69ebbcf338	0
0xaad00f5dc5a76a9fa9060cd3cdfce9bfbe0e5e81	0
0xf014581e9d5907817d5a85055793a94cb66a909e	0
0x0b5cba1bc5ec084e82fadc4e2c3a43a94660b6c4	0
0x820b5d21d1b1125b1aad51951f6e032a07caec65	1
0x7e0949df59f3e71b22c409a97887adf970fc011b	0
0x06cda3b81bcc049c6e379ab394adb851d2844c31	0
0x9a7A4ea93CE5a9129D754B7Dbeb3a3968939D6b3	0
0x8cd3bac9875b1945d1d346994723 6d8971bf3174	0
0xbc1aB7F58a95cf307Ba5Ee18CF06b7eD0120da7a	0

Berdasarkan alamat smart contract tersebut maka bisa didownload bytecode dari https://console.cloud.google.com/marketplace/product/ethereum/crypto-ethereum-blockchain. Bytecode merupakan urutan syntax berupa angka heksadesimal berisi instruksi dan data yang diproses. Contoh bytecode seperti pada tabel 4.4.

Tabel 4.4. Contoh bytecode dari Google BigQuery

Bytecode	Ponzi
0x6060604052361561006ad576000357c0100000000000000000000000000000000000000 0000000000000000000000900463ffffffff16806306fdde03146100b2578063095ea7b31 461014157806318160ddd146101d457806323b872dd1461 01fd578063313ce567146102765780637a0823114610291f57806395d89b41146102ec 578063a0712d681461037b578063a9059cbb146103b6578063dd62ed3e14610410575 b600080fd5b34156100bd57600080fd5b6100c561047c565b6040518080602001828101 3825283818151815260200191508051906020019080838360005b838110156101065 78082015181840152602081019050610ea565b50505050905090810190601f16801 561013357808203805160018360200361010000a031916815260200191505b50925050 506040518091039ff35b341561014c57600080fd5b610181600480803573ffffffffffffff ffffffffffffffffffffffff169060200190919080359060200190919050506105a565b6040 518082151515158152602001915050604051809103906f35b34156101a657600080fd5b 6101d2600480803573ffffffffffffffffffffffffffffffffffffffff169060200190919050506106 0d565b005b34156101df57600080fd5b6101e76107bb565b604051808281526020019 1505060405180910390f35b341561020857600080fd5b61025c600480803573ffffffffffff ffffffffffffffffffffffff169060200190919080359060200190919080359060200190919080 3573ffffffffffffffffffffffffffffffffffffffff169060200190919050506107c1565b604051808215151515815 260200191505060405180910390f35b34156102815760006080fd5b61028e610a42565b60 405180828152602001915050604051809103906f35b34156102aa57600080fd5b6102d6 600480803573ffffffffffffffffffffffffffffffffffffffff169060200190919050506610a48565b 604051808281526020019150506040518091039ff35b34156102f757600080fd5b6102 ff610a92565b604051808060200182810382528381815181526020019150805190602 0190808383600005b838110156103405780820151818401525b602081019050610324 565b505050509050908101906601f16801561036d5780820380516001836020036101000 0a031916815260200191505b509250505060405180910390f35b34156103865760008 0fd5b61039c600480803590602001909190505061 0b30565b604051808215151515815 26020019150506040518091039ff35b34156103c157600080fd5b6103f660048080357 3ffffffffffffffffffffffffffffffffffffffff169060200190919080359060200190919050506 0c78565b604051808215151515815260200191505060405180910390f35b341561041 b57600080fd5b610466600480803573ffffffffffffffffffffffffffffffffffffffff16906020019 09190803573ffffffffffffffffffffffffffffffffffffffff169060200190919050506 10e0b565b6 040518082815260200191505060405180910390f35b600080546001816001161561010	0

Tabel 4.2. Contoh bytecode dari Google BigQuery (lanjutan)

Bytecode	Ponzi
09190803573fffffffffffffffffffffffffffffffffffffff1690602001909190505061 0e0b565b6040518082815260200191505060405180910390f35b600080546001816001161561010002031660029004801561051257806 01f106104e75761010080835404028352916020019161051256 5b820191906000526020600020905b81548152906001019060 20018083116104f55782900360 1f168201915b5050505050815 65b6000816005600033 73ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff1681526020019081526020016000206000 8573ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff168152602001908152602001600020819055508273ffffffffffffffffffffffffffffffffffffffff163373ffffffffffffffffffffffffffffffffffffffff167f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b925846040518082815260200 ...	0

Untuk mendapatkan opcode berdasarkan alamat smart contract, maka bisa didownload dari ethereum.io dengan layanan API. Opcode yang didapat masih tercampur dengan syntax html seperti pada tabel 4.5. Maka perlu dibersihkan sebelum data diproses.

Pengambilan data bytecode dan opcode dilakukan karena keduanya adalah obyek utama dari penelitian. Adapun penggunaan smart contract yang berupa source code tidak bisa dilakukan karena hanya sedikit data smart contract dalam bentuk source code. Jika bytecode atau opcode dikonversi atau diterjemahkan balik ke source code maka terjadi banyak reduksi informasi. Apalagi pengaruh compiler dan decompiler yang berbeda mekanisme dan versinya.

Tabel 4.5. Contoh Opcode dari Ehterscan.io

Opcode	Ponzi
PUSH1 0x60 PUSH1 0x40 MSTORE CALLDATASIZE ISZERO PUSH2 0x00b9 JUMPI PUSH1 0xe0 PUSH1 0x02 EXP PUSH1 0x00 CALLDATALOAD DIV<	1

Tabel 4.3. Contoh Opcode dari Ehterscan.io (lanjutan)

Opcode	Ponzi
0x02fb0c5e DUP2 EQ PUSH2 0x03bd JUMPI DUP1 PUSH4 0xd24257c0 EQ PUSH2 0x0405 JUMPI DUP1 PUSH4 0xf71d96cb EQ PUSH2 0x040e JUMPI JUMPDEST PUSH2 0x0450 PUSH1 0x01 SLOAD PUSH1 0xff AND ISZERO PUSH2 0x0452 JUMPI PUSH1 0x03 SLOAD PUSH1 0x00 DUP1 SLOAD PUSH2 0x0457 SWAP3 SWAP1 LT PUSH2 0x0547 JUMPI PUSH1 0x07 SLOAD NUMBER GT ISZERO PUSH2 0x0521 JUMPI PUSH2 0x0547 JUMPDEST PUSH1 0x40 DUP1 MLOAD PUSH1 0x20 DUP2 ADD DUP3 MSTORE PUSH1 0x00 DUP1 DUP3 MSTORE PUSH1 0x06 DUP1 SLOAD PUSH1 0x07 SLOAD PUSH1 0xf8 PUSH1 0x02 EXP SWAP1 BLOCKHASH DUP2 DUP2 DIV DUP3 MUL SWAP2 SWAP1 SWAP2 DIV PUSH1 0xff NOT SWAP1 SWAP3 AND SWAP2 SWAP1 SWAP2 OR SWAP1 SWAP2 SSTORE PUSH1 0x03 SLOAD SWAP4 MLOAD SWAP1 SWAP4 SWAP2 SWAP3 DUP4 SWAP3 SWAP1 SWAP2 DUP4 SWAP2 DUP3 SWAP2 DUP3 SWAP2 SWAP1 DUP1 MSIZE LT PUSH2 0x0137 JUMPI POP MSIZE JUMPDEST DUP2 DUP2 MSTORE PUSH1 0x20 SWAP2 DUP3 MUL DUP2 ADD SWAP1 SWAP2 ADD PUSH1 0x40 MSTORE PUSH1 0x00 SWAP6 POP SWAP4 POP DUP5 SWAP3 POP JUMPDEST PUSH1 0x00 SLOAD PUSH1 0xff DUP7 AND LT ISZERO PUSH2 0x07a9 JUMPI PUSH1 0x00 DUP1 SLOAD DUP7 SWAP1 DUP2 LT ISZERO PUSH2 0x0002 JUMPI SWAP1 DUP1 MSTORE PUSH1 0x00 DUP1 MLOAD PUSH1 0x20 PUSH2 0x084d DUP4 CODECOPY DUP2 MLOAD SWAP2 MSTORE ADD DUP1 SLOAD SWAP1 SWAP3 POP PUSH1 0xff PUSH1 0x02 EXP PUSH1 0xa0 PUSH1 0x02 EXP SWAP1 SWAP2 DIV PUSH1 0xf8 PUSH1 0x02 EXP MUL LT DUP1 ISZERO PUSH2 0x01b1 JUMPI POP PUSH1 0x06 SLOAD PUSH1 0xff PUSH1 0x02 EXP PUSH1 0xf8 PUSH1 0x02 EXP SWAP2 SWAP1 SWAP2 MUL LT JUMPDEST DUP1 PUSH2	1

0x01e9 JUMPI POP DUP2 SLOAD PUSH1 0xff PUSH1	

Tabel 4.3. Contoh Opcode dari Ehterscan.io (lanjutan)

Opcode	Ponzi
0x02 EXP SUB SWAP1 DUP2 AND CALLER SWAP 1 SWAP2 AND EQ ISZERO PUSH2 0x0457 JUMPI PUSH1 0x01 DUP1 SLOAD PUSH1 0xff NOT AND SWAP1 SSTORE PUSH2 0x051e PUSH2 V DUP2 MUL SWAP2 DUP3 SWAP1 MSTORE DUP 5 SLOAD SWAP2 DIV PUSH1 0xa0 PUSH1 0x02 EXP MUL PUSH20 0xffffffffffffffffffffffffffffffffffffffff NOT SWAP2 SWAP1 SWAP2 AND CALLER OR PUSH21 0xff0000000000000000000000000000000000000000 NOT AND OR SWAP1 SWAP3 SSTORE POP POP NUMBER PUSH1 0x07 SSTORE POP JUMPDEST POP JUMP JUMPDEST< br>PUSH2 0x055e JUMPDEST PUSH1 0x40 MLOAD CALLER PUSH1 0x01 PUSH1 0xa0 PUSH1 ...	1

4.2. Preprocessing Data

Sebelum dataset digunakan untuk uji klasifikasi maka perlu adanya data preprocessing. Preprocessing data tersebut berupa:

1) Menghapus data yang mempunyai label error.
2) Memisahkan data bytecode per dua karakter dengan spasi. Hal ini dilakukan karena instruksi pada bytecode berupa code heksademimal 2 digit. Tetapi metode ini tidak bisa membedakan data atau instruksi, sehingga code heksadesimal yang berisi data terkadang lebih dari 2 digit.
3) Mengubah alamat akun yang berjumlah 42 karakter pada opcode dengan kata 'ADDRESS_ACCOUNT', dan hash transaksi yang berjumlah 66 karakter dengan kata 'TRANSACTION_HASH'.

4) Mengubah karakter heksademimal pada opcode selain alamat akun dan hash transaksi dengan kata 'UNKNOWN_HEX'.

Hasil akhir yang didapat adalah label ponzi, alamat smart contract, bytecode dan opcode berjumlah 2857 data yang terdiri dari 2726 data bukan ponzi dan 131 berupa ponzi. Data ini lebih sedikit dibandingkan dataset Chen dkk (2019) yang berjumlah 3794 data. Hal ini dikarenakan terdapat value label error dan beberapa tidak ditemukan di BigQuery ataupun etherscan.io dimungkinkan smart contract tersebut mengeksekusi self destruct sehingga terhapus dari semua server ethereum.

	label	bytecode_pecah	opcode_unknown_hex	opcode_del_unknown_hex
0	0	0x 60 60 60 40 52 36 15 61 00 ad 57 60 00 35 7...	PUSH1 UNKNOWN_HEX PUSH1 UNKNOWN_HEX MSTORE CAL...	PUSH1 PUSH1 MSTORE CALLDATASIZE ISZERO PUSH2 J...
1	0	0x 60 60 60 40 52 36 15 61 00 67 57 63 ff ff f...	PUSH1 UNKNOWN_HEX PUSH1 UNKNOWN_HEX MSTORE CAL...	PUSH1 PUSH1 MSTORE CALLDATASIZE ISZERO PUSH2 J...
2	0	0x 60 60 60 40 52 36 15 61 01 10 57 60 00 35 7...	PUSH1 UNKNOWN_HEX PUSH1 UNKNOWN_HEX MSTORE CAL...	PUSH1 PUSH1 MSTORE CALLDATASIZE ISZERO PUSH2 J...
3	0	0x 60 60 60 40 52 36 15 61 00 ef 57 60 00 35 7...	PUSH1 UNKNOWN_HEX PUSH1 UNKNOWN_HEX MSTORE CAL...	PUSH1 PUSH1 MSTORE CALLDATASIZE ISZERO PUSH2 J...
4	0	0x 60 60 60 40 52 36 15 61 00 5f 57 63 ff ff f...	PUSH1 UNKNOWN_HEX PUSH1 UNKNOWN_HEX MSTORE CAL...	PUSH1 PUSH1 MSTORE CALLDATASIZE ISZERO PUSH2 J...
...	...	...	...	...
2852	0	0x 60 60 60 40 52 36 15 61 00 b9 57 60 e0 60 0...	PUSH1 UNKNOWN_HEX PUSH1 UNKNOWN_HEX MSTORE CAL...	PUSH1 PUSH1 MSTORE CALLDATASIZE ISZERO PUSH2 J...
2853	1	0x 60 60 60 40 52 36 15 61 00	PUSH1 UNKNOWN_HEX PUSH1	PUSH1 PUSH1 MSTORE CALLDATASIZE

Gambar 4.8. Dataset bytecode dan opcode

4.3. Uji Klasifikasi

Proses uji klasifikasi ini digunakan beberapa algoritma untuk mengklasifikasi data set yang tersedia. Perbandingan yang dilakukan adalah dataset bytecode, opcode dengan UNKNOWN_HEX, dan opcode tanpa UNKNOWN_HEX. Setelah itu perbandingan data yang imbalanced dengan balanced menggunakan SMOTE. Kemudian implmentasi klasifikasi dengan algoritma yang berbeda-beda.

Tabel 4.6. Daftar algoritma yang digunakan untuk klasifikasi

No	Algoritma
1	LogisticRegression
2	LinearDiscriminantAnalysis
3	KneighborsClassifier
4	DecisionTreeClassifier
5	GaussianNB
6	SVC
7	AdaBoostClassifier
8	GradientBoostingClassifier
9	RandomForestClassifier
10	XGBClassifier
11	LinearSVC
12	MLPClassifier

Parameter awal yang digunakan dalam proses training metode TF-IDF adalah max_features=10000, min_df=1, ngram_range=1,5, stop_words=None. Ketika sudah ditemukan algoritma yang terbaik, maka dilakukan pengujian secara iteratif dalam hal perbandingan ngram_range. Sedangkan untuk metode ujinya menggunkan fold sebanyak 5 kali dengan nilai data diacak. Split data antara train dan test sebesar 80 dibanding 20.

Pada proses training TF-IDF data set berupa bytecode dan opcode akan di susun semacam token pada NLP, sehingga akan muncul fitur-fitur yang diprediksi

mempengaruhi akurasi deteksi. Pada penelitian ini fitur-fitur yang optimal memberi pengaruh akurasi tidak dipertimbangkan. Pola dari hasil klasifikasi, max_feature dengan jumlah 10.000 merupakan hasil optimal, jika lebih dari itu makak tidak ada penambahan hasil uji klasifikasi.

4.4. Hasil Klasifikasi

Hasil dari uji klasifikasi menggunakan F-Score adalah sebagai berikut:

Tabel 4.7 Hasil Klasifikasi dari Dataset Bytecode Imbalanced

Balancing	Algoritma	Hasil F-Score
Imbalanced	LogisticRegression	95.88
	LinearDiscriminantAnalysis	77.07
	KneighborsClassifier	97.48
	DecisionTreeClassifier	97.15
	GaussianNB	96.27
	SVC	97.13
	AdaBoostClassifier	97.70
	GradientBoostingClassifier	97.84
	RandomForestClassifier	97.79
	XGBClassifier	**97.92**
	LinearSVC	97.36
	MLPClassifier	97.86

Tabel 4.8. Hasil Klasifikasi dari Dataset Bytecode Diimbangkan dengan SMOTE

Balancing	Algoritma	Hasil F-Score
SMOTE	LogisticRegression	95.88
	LinearDiscriminantAnalysis	77.07
	KneighborsClassifier	97.48
	DecisionTreeClassifier	97.03
	GaussianNB	96.27
	SVC	97.13
	AdaBoostClassifier	97.70
	GradientBoostingClassifier	97.83
	RandomForestClassifier	97.69
	XGBClassifier	**97.92**
	LinearSVC	97.36
	MLPClassifier	97.81

Tabel 4.9. Hasil Klasifikasi dari Dataset Opcode Dengan UNKNOWN HEX

Imbalanced

Balancing	Algoritma	Hasil F-Score
Imbalanced	LogisticRegression	95.08
	LinearDiscriminantAnalysis	91.73
	KneighborsClassifier	97.17
	DecisionTreeClassifier	96.93
	GaussianNB	97.07
	SVC	96.16
	AdaBoostClassifier	97.77
	GradientBoostingClassifier	97.80
	RandomForestClassifier	97.78
	XGBClassifier	**98.04**
	LinearSVC	96.61
	MLPClassifier	97.84

Tabel 4.10. Hasil Klasifikasi dari Dataset Opcode Dengan UNKNOWN HEX diimbangkan dengan SMOTE

Balancing	Algoritma	Hasil F-Score
SMOTE	LogisticRegression	95.08
	LinearDiscriminantAnalysis	91.73
	KneighborsClassifier	97.17
	DecisionTreeClassifier	97.05
	GaussianNB	97.07
	SVC	96.16
	AdaBoostClassifier	97.77
	GradientBoostingClassifier	97.75
	RandomForestClassifier	97.84
	XGBClassifier	**98.04**
	LinearSVC	96.61
	MLPClassifier	97.98

Tabel 4.11. Hasil Klasifikasi dari Dataset Opcode TANPA UNKNOWN HEX

Imbalanced

Balancing	Algoritma	Hasil F-Score
Imbalanced	LogisticRegression	95.81
	LinearDiscriminantAnalysis	92.80
	KneighborsClassifier	97.18
	DecisionTreeClassifier	96.77
	GaussianNB	97.12
	SVC	97.11
	AdaBoostClassifier	97.75
	GradientBoostingClassifier	97.25
	RandomForestClassifier	97.86
	XGBClassifier	**98.13**
	LinearSVC	97.18
	MLPClassifier	97.89

Tabel 4.12. Hasil Klasifikasi dari Dataset Opcode TANPA UNKNOWN HEX diimbangkan dengan SMOTE

Balancing	Algoritma	Hasil F-Score
SMOTE	LogisticRegression	94.74
	LinearDiscriminantAnalysis	91.82
	KneighborsClassifier	91.65
	DecisionTreeClassifier	96.59
	GaussianNB	97.28
	SVC	97.10
	AdaBoostClassifier	97.36
	GradientBoostingClassifier	97.70
	RandomForestClassifier	**97.79**
	XGBClassifier	97.73
	LinearSVC	96.29
	MLPClassifier	97.56

Ketika menggunakan parameter awal dalam proses training metode TF-IDF berupa max_features=10000, min_df=1, ngram_range=1,5, stop_words=None ditemukan hasil yang paling optimal menggunakan algoritma XGBClassifier sebesar F-score 98.13%. Selanjutnya adalah uji coba secara iterasi dengan penggantian kombinasi ngram_range, dimana hasilnya adalah sebagai berkut:

Tabel 4.13. Perbandingan hasil klasifikasi dengan optimasi ngram_range

Balancing	Datasets	ngram_range	Hasil F-Score
Imbalanced	bytecode_pecah	2,5	98.07%
	opcode_unknown_hex	3,4	98.16%
	opcode_del_unknown_hex	1,5	98.13%

Untuk perbandingan hasil dengan penelitian-penelitian sebelumnya adalah

Tabel 4.14. Perbandingan hasil dengan penelitian sebelumnya

No	Author	Dataset	F-Score
1	Fan dkk (2021)	Smart Contract	96%
2	Liang dkk. (2021)	Smart Contract dan Transaksi	91%
3	Chen dkk. (2021)	Smart Contract	89%
4	Yu dkk. (2021)	Transaksi	89%
5	Shen, Jiang, dan Zhang (2021)	Smart Contract	88%
6	Wang dkk. (2021)	Smart Contract dan Akun	96%
7	Penelitian ini dengan metode XGBClassifier	Bytecode imbalanced	98.07%
		Bytecode diimbangkan dengan SMOTE	97.92%
		Opcode dengan UNKNOWN_HEX Imbalanced	**98.16%**
		Opcode dengan UNKNOWN_HEX diimbangkan dengan SMOTE	98.04%
		Opcode tanpa UNKNOWN_HEX imbalanced	98.13%
	Penelitian ini dengan metode RandomForestClassifier	Opcode tanpa UNKNOWN_HEX diimbangkan dengan SMOTE	97.79%

Gambar 4.9. Alur pengujian dari dataset sampai dengan hasil akhir

Dalam proses pencegahan penipuan skema ponzi maka model yang dihasilkan dari penelitian ini di simpan dalam bentuk sesuai dengan software atau aplikasi tujuan. Pada umumnya format model menggunakan ektensi *dat. Model ini tidak bisa dijalankan tanpa adanya matrix pembobotan TF-IDF yang telah tergenerate. Sehingga jika ada smart contract yang akan dideteksi apakah mengandung skema ponzi atau tidak, maka smart contract tersebut diambil opcodenya. Opcode yang didapat dimodifikasi dengan cara mengubah alamat akun dengan string ACCOUNT_ADDRESS, transaksi hash dengan string TRANSACTION_HASH, dan data heksadesimal selain dua poin sebelumnya diganti dengan string UNKNOWN_HEX. Selanjutnya opcode tersebut diberi pembobotan menggunakan matrix yang sudah tergenerate dari penelitian ini. Nilai bobot dideteksi menggunakan model dari penelitian ini. Hasil yang muncul adalah nilai status ponzi atau tidak.

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil uji klasifikasi maka dapat ditarik kesimpulan bahwa algoritma yang paling optimal untuk dataset bytecode dan dataset opcode dengan UNKNOWN_HEX adalah XGBClassifier. Adapun untuk dataset opcode tanpa UNKONWN_HEX algoritma yang paling optimal adalah XGBClassifier dan RandomForestClassifier.

Berdasarkan dataset yang diuji antara bytecode dan opcode, maka hasil yang lebih baik adalah dataset opcode. Dimana hasil maksimal dari bytecode adalah 98.07%, sedangkan dataset opcode bisa mencapai hasil 98.16%. Adapun parameter yang optimal dalam proses pembobotan TF-IDF adalah max_features=10000 dan ngram_range=(3,4).

Hasil optimal pada penelitian ini sudah lebih baik daripada dengan penelitian sebelumnya (Fan dkk (2021). Penelitian sebelumnya bisa mencapai hasil F-Score 96%, sedangkan penelitian ini mencapai hasil 98.16%. Penelitian sebelumnya menggunakan Bag of Word untuk pembobotan syntax opcode, sedangkan penelitian ini menggunkanan TF-IDF. Selain itu penelitian sebelumnya memberlakukan stop word, sedangkan penelitian ini tidak menggunakan stop word dikarenakan sintak opcode tidak bisa dianggap kalimat yang sering muncul dan tidak bermakna. Padahal posisi kemunculan sintak tersebut sangat memberi pengaruh pada pembobotan.

Jika dibandingkan penelitian sebelumnya yang lain yaitu Liang dkk (2021) mempunyai hasil 91% menggunakan dataset smart contract dan transaksi. Ada juga Chen dkk (2021) dengan hasil 89% mengunakan smart contract berupa source code bukan bytecode ataupun opcode. Selanjutnya Yu dkk (2021) menggunakan dataset transaksi saja mendapatkan hasil 89%. Shen, Jiang dan Zhang (2021) mendapatkan hasil 88% berdsarkan Smart Contract. Begitu pula Wang dkk (2021) menggunakan smart contract dan human account mendapatkan hasil 96%.

Pada penelitian ini masih sebatas pada pembentukan model, belum sampai pada percobaan deteksi seluruh smart contract yang berjalan di seluruh Blockchain Ethereum.

5.2. Saran

Untuk penelitian selanjutnya ada beberapa hal yang bisa diperbaiki dari penelitian ini:

a. Matrik pembobotan TF-IDF masih bisa diperkecil, sehingga komputasi juga semkin ringan.
b. Dataset yang tidak imbang dan jumlahnya sedikit bisa diperbaiki lagi dengan dataset yang lebih banyak dan imbang.
c. Perlu adanya ekstraksi fitur yang memberi pengaruh besar pada hasil klasifikasi. Karena pada penelitian ini semua fitur dianggap sama.
d. Mengimplementasikan model dalam data seluruh smart contract yang berjalan.

DAFTAR PUSTAKA

PUSTAKA BUKU

Bashir, Imran. 2020. Mastering Blockchain: A Deep Dive into Distributed Ledgers, Consensus Protocols, Smart Contracts, DApps, Cryptocurrencies, Ethereum, and More.

Grincalaitis, Merunas. 2019. MASTERING ETHEREUM: Implement Advanced Blockchain Applications Using Ethereum-Supported Tools, ... Services, and Protocols. PACKT Publishing Limited.

SEC. Definition of Ponzi scheme from SEC, Jul. 2019.

Jurafsky, D. (2007). Speech and language processing. Prentice Hall.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Namasudra, Suyel, Ganesh Chandra Deka, Prashant Johri, Mohammad Hosseinpour, dan Amir H. Gandomi. 2021. "The Revolution of Blockchain: State-of-the-Art and Research Challenges." Archives of Computational Methods in Engineering 28(3):1497–1515. doi: 10.1007/s11831-020-09426-0.

Ferretti, Stefano, dan Gabriele D'Angelo. 2020. "On the Ethereum Blockchain Structure: A Complex Networks Theory Perspective." Concurrency and Computation: Practice and Experience 32(12). doi: 10.1002/cpe.5493.

Tsankov, Petar, Andrei Dan, Dana Drachsler Cohen, Arthur Gervais, Florian Buenzli, dan Martin Vechev. 2018. "Securify: Practical Security Analysis of Smart Contracts." ArXiv:1806.01143 [Cs].

Savelyev, Alexander. 2017. "Contract Law 2.0: 'Smart' Contracts as the Beginning of the End of Classic Contract Law." Information & Communications Technology Law 26(2):116–34. doi: 10.1080/13600834.2017.1301036.

Bartoletti, Massimo, Salvatore Carta, Tiziana Cimoli, dan Roberto Saia. 2019. "Dissecting Ponzi Schemes on Ethereum: Identification, Analysis, and Impact." Future Generation Computer Systems 102:259–77. doi: 10.1016/j.future.2019.08.014.

Atzei, Nicola, Massimo Bartoletti, dan Tiziana Cimoli. 2017. "A Survey of Attacks on Ethereum Smart Contracts (SoK)." Hlm. 164–86 dalam Principles of Security and Trust. Vol. 10204, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg.

Fan, Shuhui, Shaojing Fu, Haoran Xu, dan Xiaochun Cheng. 2021. "Al-SPSD: Anti-Leakage Smart Ponzi Schemes Detection in Blockchain." Information Processing & Management 58(4):102587. doi: 10.1016/j.ipm.2021.102587.

Liang, Yuzhi, Weijing Wu, Kai Lei, dan Feiyang Wang. 2021. "Data-Driven Smart Ponzi Scheme Detection." ArXiv:2108.09305 [Cs].

Chen, Yizhou, Heng Dai, Xiao Yu, Wenhua Hu, Zhiwen Xie, dan Cheng Tan. 2021. "Improving Ponzi Scheme Contract Detection Using Multi-Channel TextCNN and Transformer." Sensors 21(19):6417. doi: 10.3390/s21196417.

Yu, Shanqing, Jie Jin, Yunyi Xie, Jie Shen, dan Qi Xuan. 2021. "Ponzi Scheme Detection in EthereumTransaction Network." ArXiv:2104.08456 [Cs].

Shen, Xiajiong, Shuaimin Jiang, dan Lei Zhang. 2021. "Mining Bytecode Features of Smart Contracts to Detect Ponzi Scheme on Blockchain." Computer Modeling in Engineering & Sciences 127(3):1069–85. doi: 10.32604/cmes.2021.015736.

Wang, Lei, Hao Cheng, Zibin Zheng, Aijun Yang, dan Xiaohu Zhu. 2021. "Ponzi Scheme Detection via Oversampling-Based Long Short-Term Memory for Smart Contracts." Knowledge-Based Systems 228:107312. doi: 10.1016/j.knosys.2021.107312.

Suthaharan, Shan. 2016. Machine Learning Models and Algorithms for Big Data Classification. Vol. 36. Boston, MA: Springer US.

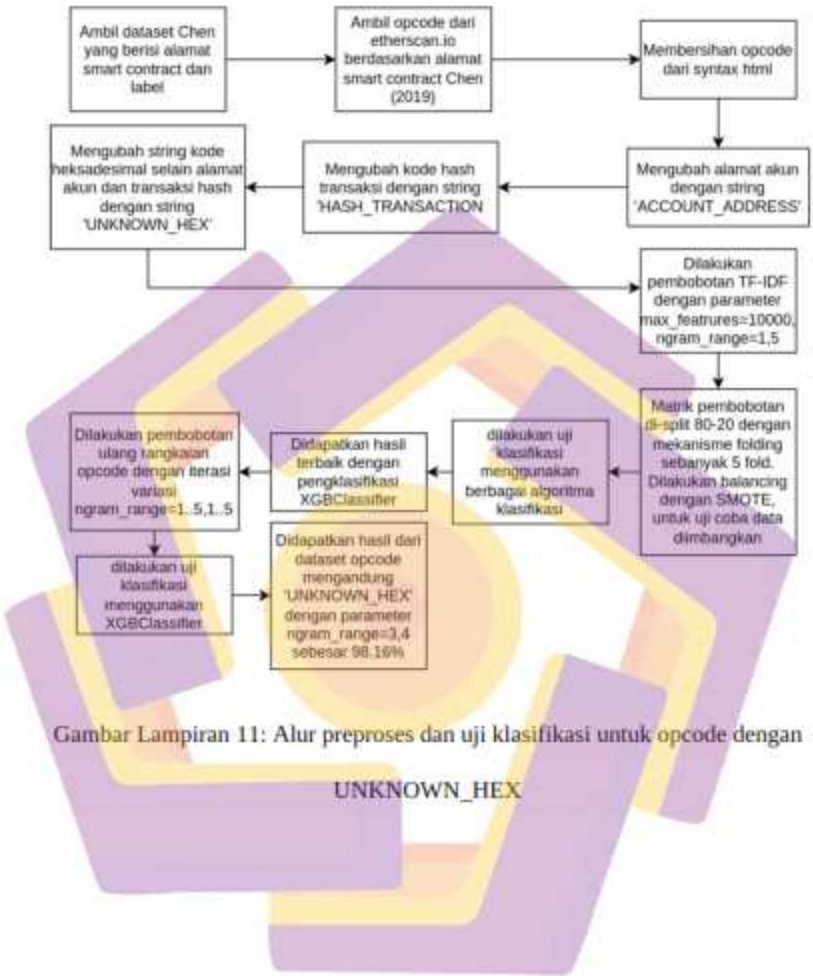
Chen, Weili, Zibin Zheng, Edith C. H. Ngai, Peilin Zheng, dan Yuren Zhou. 2019. "Exploiting Blockchain Data to Detect Smart Ponzi Schemes on Ethereum." IEEE Access 7:37575–86. doi: 10.1109/ACCESS.2019.2905769.

PUSTAKA ELEKTRONIK

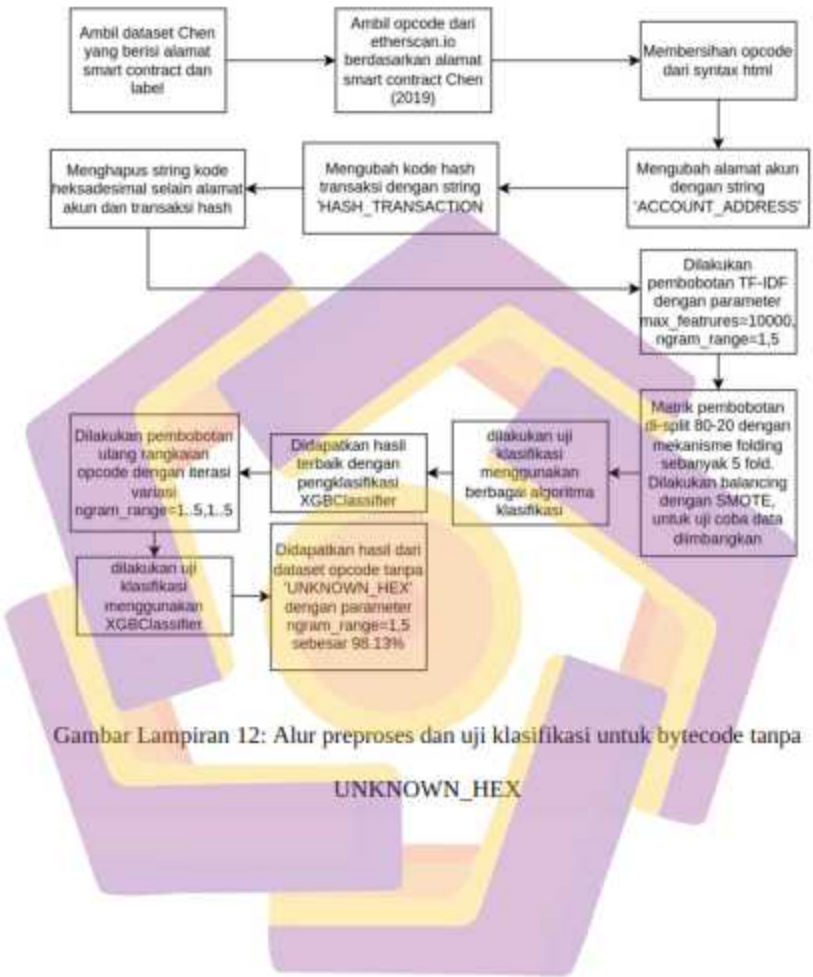
CoinMarketCap.com. (2021). Bitcoin price, charts, market cap, and other metrics | CoinMarketCap.

LAMPIRAN

Gambar Lampiran 10: Alur preproses dan uji klasifikasi untuk bytecode



Gambar Lampiran 11: Alur preproses dan uji klasifikasi untuk opcode dengan UNKNOWN_HEX



Gambar Lampiran 12: Alur preproses dan uji klasifikasi untuk bytecode tanpa UNKNOWN_HEX