

Analisis Graf untuk Mengungkap Distribusi Aktual Token di Ekosistem Blockchain Ethereum

Juan Alfred Widjaya - 13522073¹
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹13522073@std.stei.itb.ac.id

Abstrak—Distribusi token yang merata dan terdesentralisasi memiliki dampak besar pada dinamika pasar kripto. Dalam analisis ini, data pemegang token dikumpulkan dan digunakan untuk membangun graf berbobot yang menggambarkan transaksi antar pemegang token. Algoritma Louvain diterapkan untuk mengidentifikasi komunitas dalam jaringan token, mengungkapkan bahwa sebagian besar pemegang token bertindak secara independen. Hasil analisis ini memperkuat konsep desentralisasi dalam ekosistem kripto, menunjukkan bahwa pasar tidak didominasi oleh pemegang besar yang dapat mempengaruhi harga dan stabilitas pasar. Pemahaman ini penting untuk mengukur risiko dan potensi investasi dalam kripto.

Kata kunci—Algoritma Louvain, Distribusi, Graf, Token Ethereum

I. PENDAHULUAN

Blockchain Ethereum telah memainkan peran yang krusial dalam mengubah lanskap ekonomi global dengan memungkinkan pengembangan berbagai jenis kontrak pintar yang berjalan secara otomatis dan aman. Keunggulan utama Ethereum tidak hanya terletak pada kemampuannya untuk mendukung kontrak pintar, tetapi juga pada ekosistem yang kaya dengan berbagai jenis token kripto yang diterbitkan di atasnya. Token-token ini mencakup mata uang kripto seperti Ether (ETH), representasi digital dari aset-aset fisik, token utilitas yang memberikan akses ke layanan atau produk tertentu, serta token NFT (Non-Fungible Token) yang mewakili kepemilikan unik atas aset digital.

Salah satu aspek penting yang harus dipahami dalam ekosistem kripto adalah distribusi token. Distribusi token yang merata dan terdesentralisasi memiliki dampak signifikan pada dinamika pasar kripto secara keseluruhan. Ketika distribusi terpusat pada pemegang token besar, hal ini dapat memberikan mereka kekuatan untuk mempengaruhi harga dan stabilitas pasar dengan mudah. Selain itu, fenomena di mana satu individu atau entitas memegang sejumlah besar token yang sama di berbagai dompet (*wallet*) memberikan kesan bahwa token tersebut memiliki pemegang yang banyak, tetapi sebenarnya terpusat pada satu entitas, yang dapat mengancam sifat terdesentralisasi yang diidamkan dalam teknologi blockchain.

Pemahaman yang mendalam tentang distribusi aktual token

di Ethereum memiliki implikasi yang luas dalam ekosistem kripto. Bagi para investor, distribusi yang terdesentralisasi menjadi kunci dalam mengukur risiko dan potensi investasi, karena distribusi yang merata dapat mengurangi potensi manipulasi harga oleh pemegang besar. Analisis distribusi token juga berperan dalam pengukuran risiko, membantu mengidentifikasi potensi kerentanan terhadap pengaruh signifikan oleh pemegang besar yang dapat mempengaruhi stabilitas pasar. Terakhir, distribusi yang merata mencerminkan salah satu tujuan fundamental dari teknologi kripto, yaitu menciptakan sistem keuangan yang lebih terdesentralisasi dan merata, mencerminkan prinsip dasar desentralisasi yang menjadi landasan ekosistem kripto.

Dalam makalah ini, penulis akan menjalani analisis graf yang mendalam untuk mengungkap distribusi aktual token di ekosistem Ethereum. Dalam analisis ini, penulis akan menggunakan teknik klusterisasi dengan algoritma Louvain untuk mengidentifikasi kelompok atau komunitas dalam jaringan token, yang dapat memberikan wawasan tentang bagaimana token-token tertentu terdistribusi di antara pemegangnya.

II. STUDI LITERATUR

A. Graf dan Definisinya

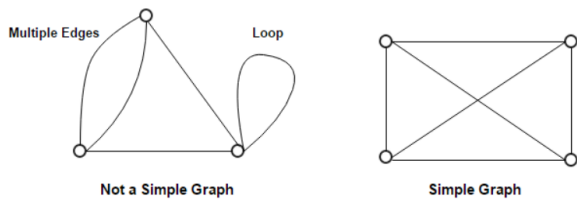
Graf adalah suatu struktur yang terbentuk dari simpul dan sisi. Graf didefinisikan sebagai $G = (V, E)$, dengan V adalah himpunan yang berisi simpul (*vertices* atau *node*) dan E adalah himpunan yang berisi sisi (*edges*). Sisi merepresentasikan hubungan antara simpul-simpul pada graf[1]

$$V = \{v_1, v_2, v_3, \dots, v_n\}$$

$$E = \{e_1, e_2, e_3, \dots, e_n\}$$

Graf dapat dibagi menjadi dua berdasarkan ada atau tidak adanya gelang pada suatu sisi, yaitu

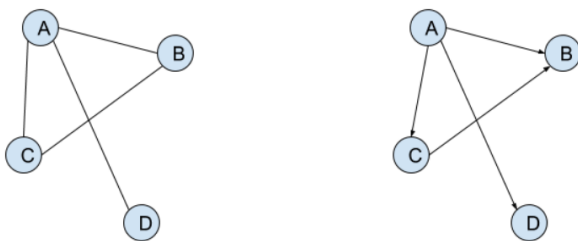
1. Graf sederhana (*simple graph*): graf yang tidak memiliki gelang dan sisi ganda.
2. Graf tak sederhana (*unsimple graph*): graf yang memiliki gelang atau sisi ganda.



Gambar 1 (a) Graf Tak Sederhana (b) Graf Sederhana
(Sumber: <https://www.javatpoint.com/graph-theory-types-of-graphs> diakses pada 8 Desember 2023 pukul 22.09 WIB)

Berdasarkan orientasi arah pada sisinya, graf dapat dibagi menjadi dua, yaitu

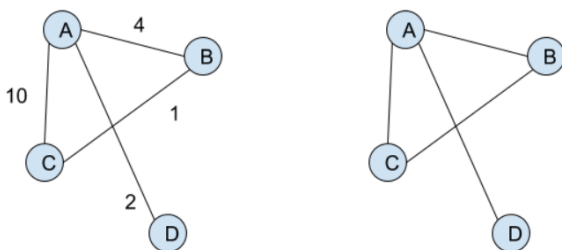
1. Graf tak berarah (*undirected graph*): graf yang tidak memiliki orientasi arah pada sisinya.
2. Graf berarah (*directed graph*): graf yang memiliki orientasi arah pada sisinya.



Gambar 2 (a) Graf Tak Berarah (b) Graf Berarah
(Sumber: <https://nycomdories.com/intro-to-graphs/> diakses pada 8 Desember 2023 pukul 22.13 WIB)

Berdasarkan ada atau tidaknya bobot pada sisinya, graf dapat dibagi menjadi dua, yaitu

1. Graf tak berbobot (*unweighted graph*): graf yang tidak memiliki bobot pada sisinya.
2. Graf berbobot (*weighted graph*): graf yang memiliki bobot pada sisinya.



Gambar 3 (a) Graf Berbobot (b) Graf Tak Berbobot
(Sumber: <https://nycomdories.com/intro-to-graphs/> diakses pada 8 Desember 2023 pukul 22.15 WIB)

B. Terminologi Graf

1. Ketetangaan (*Adjacent*): Dua buah simpul disebut bertetangaan jika dihubungkan oleh setidaknya satu sisi.
2. Bersisian (*Incidency*): Sebuah sisi dikatakan berisikan dengan suatu simpul jika sisi tersebut terhubung dengan simpul tersebut.
3. Simpul Terpencil (*Isolated Vertex*): Sebuah simpul disebut terpencil jika simpul tersebut tidak memiliki

sisi yang berisikan dengannya.

4. Graf Kosong (*Null Graph* atau *Empty Graph*): Sebuah graf disebut kosong jika graf tersebut tidak memiliki sisi.
5. Derajat (*Degree*): Derajat suatu simpul adalah jumlah sisi yang berisikan dengan simpul tersebut.
6. Lintasan (*Path*): Sebuah lintasan adalah sekumpulan sisi dan simpul yang tersusun berurutan. Sisi-sisi dalam lintasan saling terhubung secara berurutan, dan terdapat simpul di antara sisi-sisi tersebut.
7. Siklus (*Cycle*) atau Sirkuit (*Circuit*): Sebuah siklus adalah jenis khusus dari lintasan yang berawal dan berakhir di simpul yang sama.
8. Keterhubungan (*Connected*): Dua buah simpul disebut terhubung jika terdapat setidaknya satu lintasan yang menghubungkan kedua simpul tersebut.
9. Upagraf (*Subgraph*) dan Komplemen Upagraf (*Complement Subgraph*): Sebuah upagraf adalah sebagian dari suatu graf yang lebih besar. Komplemen upagraf adalah upagraf yang melengkapi upagraf yang telah ada sehingga membentuk graf yang lebih besar.
10. Upagraf Merentang (*Spanning Subgraph*): Sebuah upagraf merentang adalah upagraf yang memiliki semua simpul yang ada dalam graf yang lebih besar.
11. Cut-set (*Himpunan Potong*): Cut-set adalah himpunan sisi dalam suatu graf yang jika dipotong akan memisahkan graf tersebut menjadi dua komponen yang terpisah.

C. Ethereum dan Token ERC-20

Ethereum adalah platform komputasi terdesentralisasi, sumber terbuka, dan terdistribusi yang memungkinkan pembuatan kontrak pintar dan aplikasi terdesentralisasi, yang juga dikenal sebagai dapps (*decentralized application*). Kontrak pintar adalah protokol komputer yang memfasilitasi, memverifikasi, atau menegakkan perundingan dan pelaksanaan suatu kesepakatan. Sebagai contoh, kontrak pintar dapat digunakan untuk mewakili sebuah kontrak hukum yang mengeksplifikasikan logika dari klausa-klausa kontraktual atau sebuah kontrak keuangan yang menentukan tanggung jawab dari pihak-pihak yang terlibat dan aliran otomatis dari nilai [2].

ERC-20 adalah standar teknis untuk token yang dapat dipertukarkan yang diciptakan menggunakan blockchain Ethereum. Token yang dapat dipertukarkan ini dapat digantikan dengan token lain. ERC-20 memungkinkan pengembang untuk membuat token yang dapat digunakan dengan produk dan layanan lain yang memanfaatkan kontrak pintar. Token-token ini adalah representasi dari suatu aset, hak, kepemilikan, akses, mata uang kripto, atau hal lain yang tidak unik pada dirinya sendiri tetapi dapat ditransfer [3]. Tiap token pada blockchain Ethereum masing - masing memiliki alamat kontrak yang unik.

D. Alamat Ethereum

Alamat Ethereum adalah representasi dari Entitas Pemilik Eksternal (*Externally Owned Address/EOA*) atau kontrak yang dapat menerima (alamat tujuan) atau mengirim (alamat

sumber) transaksi dalam blockchain. Terdapat dua jenis alamat: Alamat Pemilik Eksternal (*Externally Owned Address/EOA*) yang pada dasarnya adalah alamat dompet, dan Alamat Kontrak (*Contract Address*) [4].

Alamat Pemilik Eksternal (*Externally Owned Address/EOA*), juga dikenal sebagai Alamat Dompet, mengacu pada akun publik yang menyimpan dana Anda dan hanya dapat diakses oleh pasangan kunci pribadi. Sebagai contoh, alamat Ethereum terdiri dari 42 karakter heksadesimal yang berasal dari 20 byte terakhir dari kunci publik yang mengendalikan akun dengan awalan "0x" di depannya, seperti: 0x71C7656EC7ab88b098defB751B7401B5f6d8976F.

Alamat kontrak (*contract address*) mengacu pada alamat yang menjadi tuan rumah bagi sejumlah kode dalam blockchain Ethereum yang menjalankan fungsi-fungsi tertentu. Fungsi-fungsi dari alamat kontrak ini dieksekusi saat ada transaksi dengan data input terkait (interaksi kontrak) yang dikirimkan ke alamat tersebut. Alamat Pemilik Eksternal maupun Alamat Kontrak memiliki format yang sama, yaitu 42 karakter heksadesimal.

E. Algoritma Klasterisasi Louvain

Algoritma Louvain adalah algoritma klasterisasi yang digunakan untuk mengidentifikasi komunitas dalam jaringan atau graf. Tujuan utama dari algoritma ini adalah membagi *node-node* dalam graf ke dalam kelompok atau komunitas yang saling terkait erat. Algoritma Louvain tidak memerlukan informasi sebelumnya tentang jumlah komunitas atau ukuran mereka sebelum dieksekusi. Algoritma ini terbagi dalam 2 tahap: Optimisasi Modularitas dan Agregasi Komunitas [5]. Setelah tahap pertama selesai, tahap kedua akan dilanjutkan. Kedua tahap ini akan dieksekusi sampai tidak ada lagi perubahan dalam jaringan dan mencapai modularitas maksimum. Metode ini merupakan metode optimisasi yang rakus dan memiliki kompleksitas waktu $O(n * \log(n))$ dimana n adalah jumlah node dalam jaringan [6]. Berikut adalah tahapan dan langkah utama algoritma Louvain:

Tahap 1

1. Inisialisasi Komunitas: Pada awalnya, setiap *node* dalam graf dianggap sebagai komunitas terpisah, yang berarti setiap *node* menjadi satu kelompok. Ini berarti jika kita memiliki N *node*, kita juga memiliki N komunitas pada awalnya.
2. Optimasi Modularitas Per *Node*: Untuk setiap *node* i , algoritma mengukur dampak terhadap modularitas jika *node* ini dipindahkan ke komunitas tetangganya. Ini dilakukan dengan menghitung perubahan nilai modularitas, ΔQ , menggunakan rumus berikut:

$$\Delta Q = \left[\frac{\sum_{in} + 2k_{i,in}}{2m} \right] - \left[\frac{\sum_{tot} + k_i}{2m} \right]^2 - \left[\frac{\sum_{in}}{2m} \right]^2 - \left[\frac{\sum_{tot}}{2m} \right]^2 + \left[\frac{k_i}{2m} \right]^2$$

Dimana,

- \sum_{in} adalah jumlah bobot sisi di dalam komunitas yang *node* i akan bergabung.
- $k_{i,in}$ adalah jumlah bobot sisi antara *node* i

dan *node* lain dalam komunitas yang dipertimbangkan.

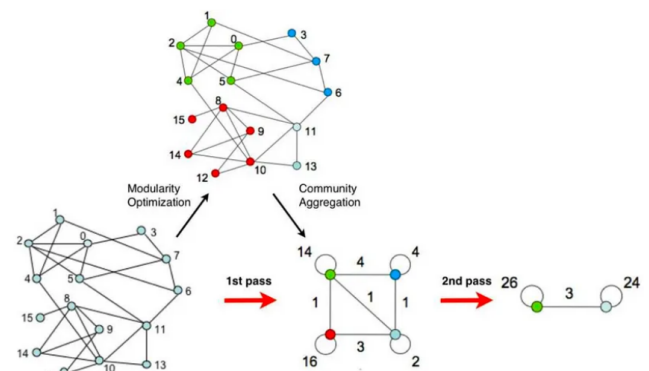
- \sum_{tot} adalah jumlah total bobot sisi yang terhubung ke komunitas yang dipertimbangkan.
 - k_i adalah jumlah bobot sisi yang terhubung ke *node* i .
 - m adalah jumlah bobot semua sisi dalam jaringan.
3. Penempatan Komunitas: *Node* i kemudian ditempatkan ke dalam komunitas yang memberikan peningkatan modularitas maksimal. Jika tidak ada peningkatan yang bisa dicapai, *node* tetap dalam komunitas aslinya.
 4. Ulangi untuk Semua *Node*: Proses di atas diulangi untuk semua *node* dalam jaringan berulang kali sampai tidak ada lagi peningkatan modularitas yang mungkin, yang menandakan bahwa sebuah plato atau maksimum lokal dari modularitas telah tercapai.

Tahap 2

Setelah tidak ada lagi peningkatan modularitas yang dapat dicapai pada tahap 1, algoritma berpindah ke tahap kedua.

1. Pembentukan Jaringan Baru: Komunitas yang terbentuk pada tahap 1 kini dijadikan sebagai *node* di jaringan baru.
2. Penetapan Bobot Sisi Baru: Sisi antara *node-node* baru ini diberi bobot yang sesuai dengan jumlah total bobot sisi antara *node-node* di komunitas-komunitas yang ada di dalam jaringan asli.
3. Iterasi: Dengan jaringan yang telah dikompres ini, algoritma kembali ke tahap 1 dan proses diulangi. Tahapan ini berlangsung hingga tidak ada lagi peningkatan signifikan terhadap modularitas yang bisa dihasilkan dari penggabungan komunitas lebih lanjut.

Proses ini berulang hingga modularitas jaringan tidak dapat lagi ditingkatkan. Pada akhirnya, kita mendapatkan jaringan yang terbagi dalam komunitas-komunitas yang mana di dalamnya terdapat *node-node* yang lebih padat terhubung satu sama lain dibandingkan dengan *node* di komunitas lain.



Gambar 4 Ilustrasi Algoritma Louvain

(Sumber: <https://towardsdatascience.com/louvain-algorithm-93fde589f58c> diakses pada 9 Desember 2023 pukul 01.35 WIB)

III. METODE

A. Pengumpulan Data dan Identifikasi Alamat

Untuk melakukan analisis distribusi suatu token, data-data *holder* dari token tersebut perlu dikumpulkan terlebih dahulu. Pada makalah ini, analisis distribusi akan dilakukan pada token PIRB (pirb.tech) dengan alamat kontrak token yakni `0xC05C3aF62f5F3663f2C415473Aa4D5AD69200033`.

Pengumpulan data-data dilakukan dengan menggunakan API dari Covalent. Data pertama yang perlu dikumpulkan adalah data *holder* atau seluruh alamat yang memiliki atau memegang token tersebut. Berikut merupakan implementasinya dalam kode python.

```
from API_KEY import *
import requests
from requests.auth import HTTPBasicAuth
import json
from concurrent.futures import ThreadPoolExecutor
from ratelimit import limits, sleep_and_retry

@sleep_and_retry
@limits(calls=5, period=1)
def fetch_transactions(holder_addr):
    global chainName, task_n, total_holders, contract_addr_list
    try:
        url_addr =
f"https://api.covalenthq.com/v1/{chainName}/address/{holder_addr}/tran
sactions_v3/"
        response = requests.get(url_addr, headers=headers, auth=basic)
        response_json = response.json()
        list_addr = {}
        for tx in response_json['data']['items']:
            address = tx['to_address']
            if address in holders and address != holder_addr:
                list_addr[address] = list_addr.get(address, 0) + 1
            api_url =
f"https://api.etherscan.io/api?module=contract&action=getsourcecode&ad
dress={holder_addr}&apikey={ETHSCAN_APIKEY}"
            response = requests.get(api_url)
            data = response.json()
            if "result" in data and data["result"] and
data["result"][0]['ContractName'] != '':
                contract_addr_list.append({'contract_addr': holder_addr,
'contract_name': data["result"][0]['ContractName']})
                task_n += 1
                print(f"TASK {task_n}/{total_holders} DONE: Get Tx from
{holder_addr}")
                return {'holder_addr': holder_addr, 'recent_tx_freq':
list_addr}
            except Exception as e:
                print(f"TASK {task_n}/{total_holders} FAIL: Get Tx from
{holder_addr}")
                fetch_transactions(holder_addr)

contract_addr_list = []
chainName = "eth-mainnet"
url_token =
f"https://api.covalenthq.com/v1/{chainName}/tokens/{TOKEN_ADDRESS}/tok
en_holders_v2/?page-size=1000"
headers = {
    "accept": "application/json",
}
basic = HTTPBasicAuth(COVALENT_APIKEY, '')
response = requests.get(url_token, headers=headers, auth=basic)
response_json = response.json()

ticker_symbol =
response_json['data']['items'][0]['contract_ticker_symbol']

holders = [holder['address'] for holder in
response_json['data']['items']]
total_holders = len(holders)
print(f"TASK DONE: Get Holders of Token {ticker_symbol}")
print(f"TOTAL HOLDERS: {total_holders}")
task_n = 0
with ThreadPoolExecutor(max_workers=8) as executor:
    data = list(executor.map(fetch_transactions, holders))

export = {'data' : data, 'token_symbol': ticker_symbol}
with open('holders.json', 'w') as file:
    json.dump(export, file)

with open('contract_addr.json', 'w') as file:
```

```
json.dump(contract_addr_list, file)
```

Seluruh kode terdapat pada *repository* GitHub (https://github.com/juanaw6/erc20token_distribution_graph).

Dalam kode yang telah disajikan, data terkait pemegang token secara keseluruhan berhasil diakses. Tahap selanjutnya yang esensial adalah pengumpulan data transaksi dari pemegang token tersebut. Proses ini merupakan langkah krusial untuk menganalisis koneksi antar alamat pemegang token. Koneksi tersebut didefinisikan melalui keberadaan transaksi antara pemegang. Transaksi ini berperan penting dalam proses pembentukan komunitas. Faktor penting yang perlu diperhatikan adalah kemungkinan bahwa satu individu mungkin memiliki lebih dari satu dompet digital dengan alamat yang berbeda. Akibatnya, sering terjadi transaksi yang frekuensinya tinggi antara dua alamat yang sejatinya dimiliki oleh individu yang sama. Situasi ini dapat menciptakan ilusi adanya jumlah pemegang token yang lebih besar dari kenyataannya.

Selain itu, identifikasi alamat juga menjadi aspek penting. Identifikasi ini bertujuan untuk membedakan apakah suatu alamat merupakan alamat pemilik eksternal atau alamat kontrak pintar. Alamat kontrak pintar dapat memegang token yang bersangkutan dan seringkali berkaitan dengan aspek-aspek seperti *treasury*, *vault*, atau *pool* dan *router swap* dari token tersebut. Alamat kontrak ini biasanya sering melakukan transaksi dengan alamat pemegang lain, dan dalam pembentukan komunitas, keberadaan alamat kontrak dapat mengganggu perhitungan dengan membentuk komunitas tersendiri, yang pada akhirnya dapat merusak analisis pemegang aktual. Dalam kode yang ada, pengidentifikasian apakah suatu alamat merupakan alamat kontrak pintar atau tidak telah dilakukan dengan menggunakan API dari Etherscan.

B. Pembuatan Graf dan Penerapan Klusterisasi

Graf berbobot yang akan dibangun dalam konteks ini berfokus pada representasi transaksi yang berkaitan dengan sebuah token kripto. Di dalam graf ini, simpul utama adalah token tersebut. Sekitar simpul utama ini, terdapat berbagai simpul lain yang merepresentasikan alamat-alamat pemegang token. Alamat-alamat ini, dalam konteks graf, dihubungkan ke simpul utama melalui sisi-sisi. Sisi ini memiliki peranan khusus, yaitu menandakan kepemilikan token oleh alamat yang bersangkutan. Oleh karena itu, sisi-sisi ini diberi bobot nol. Pemberian bobot nol ini sangat penting karena ia menghindari terbentuknya kelompok atau klaster yang tidak representatif. Artinya, dengan bobot nol, graf tidak akan menginterpretasikan bahwa alamat pemegang token memiliki hubungan langsung satu sama lain hanya karena mereka berbagi kepemilikan token yang sama.

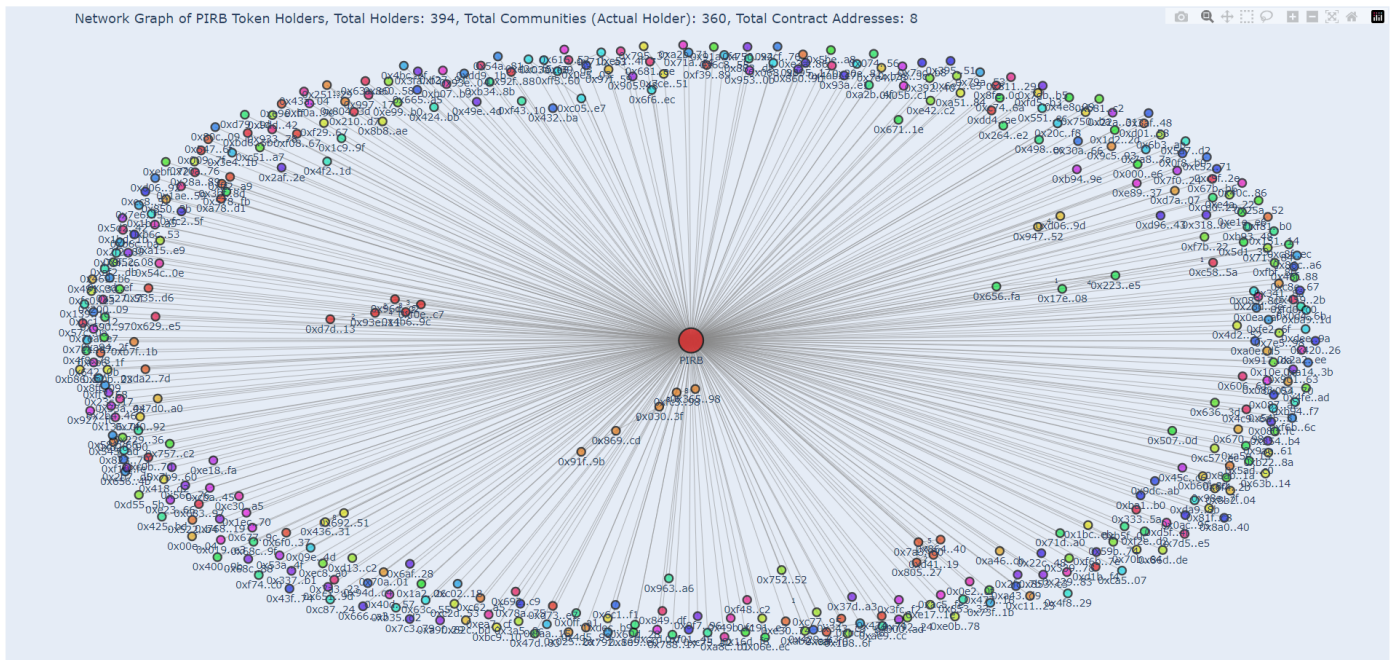
Dalam kata lain, dengan bobot nol ini, sisi antara alamat pemegang dan token utama hanya menunjukkan kepemilikan dan tidak memberikan implikasi lain mengenai hubungan antar alamat pemegang. Ini menghindari terjadinya pengelompokan yang salah dalam analisis komunitas. Pengelompokan atau klusterisasi yang sebenarnya hanya akan terbentuk berdasarkan adanya transaksi yang terjadi antara para pemegang token, yang ditandai dengan sisi berbobot lebih dari nol. Dengan

demikian, graf ini dapat dengan akurat menggambarkan bukan hanya siapa yang memiliki token, tetapi juga bagaimana mereka berinteraksi satu sama lain melalui transaksi.

Setelah pembentukan graf, algoritma klusterisasi dengan metode Louvain diimplementasikan. Algoritma ini bertujuan untuk mengidentifikasi komunitas di mana simpul-simpul terikat erat dan memiliki hubungan yang kuat. Setiap komunitas yang teridentifikasi akan diwarnai dengan warna yang memiliki keterhubungan internal yang tinggi sambil meminimalkan keterhubungan eksternal antar-komunitas.

yang berbeda-beda untuk memudahkan proses visualisasi dan pemahaman terhadap komunitas yang terbentuk.

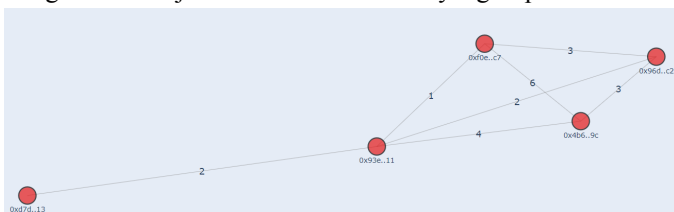
Untuk efisiensi waktu eksekusi, algoritma Louvain yang digunakan merupakan versi yang tersedia dari pustaka Python-louvain. Algoritma ini terkenal akan efisiensinya dalam menemukan komunitas dalam jaringan besar dengan cara mengoptimalkan modularitas secara iteratif. Proses ini melibatkan pengelompokan simpul-simpul ke dalam komunitas. Pendekatan ini memungkinkan penemuan struktur komunitas yang lebih akurat dan representatif dari pola interaksi dalam jaringan.



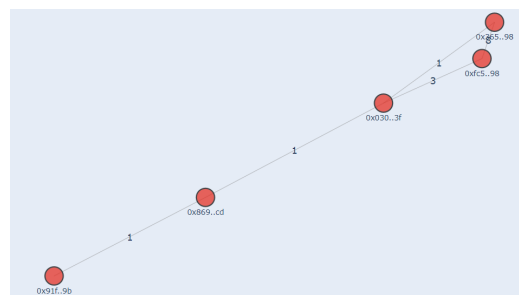
Gambar 5 Graf Berbobot Pemegang Token PIRB yang Sudah Diwarnai Sesuai Komunitas (Sumber: Arsip Pribadi)

C. Pembahasan

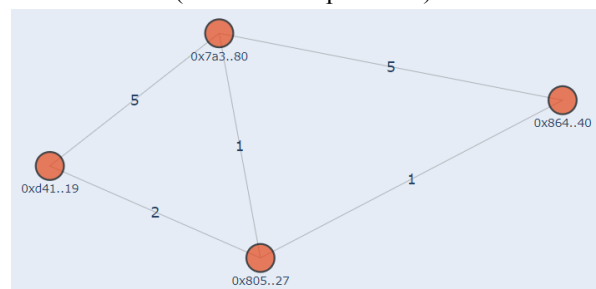
Analisis klusterisasi dalam konteks graf berbobot ini mengungkapkan sebuah fenomena menarik terkait distribusi kepemilikan token kripto. Dari total 394 alamat pemegang token, setelah dilakukan proses klusterisasi dengan metode Louvain, teridentifikasi adanya 360 komunitas yang terpisah. Fakta ini menandakan bahwa sebagian besar alamat pemegang token memiliki keterikatan yang sangat tinggi satu sama lain, menghasilkan sejumlah besar komunitas yang terpisah.



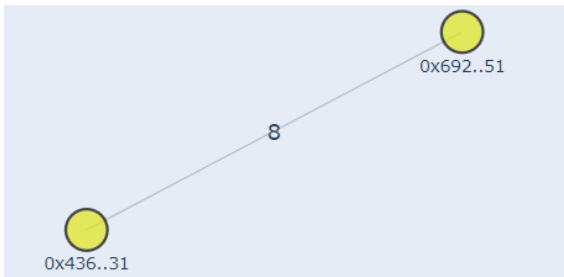
Gambar 6 Salah Satu Komunitas Besar yang Terbentuk (Sumber: Arsip Pribadi)



Gambar 7 Komunitas Lima Simpul yang Terbentuk (Sumber: Arsip Pribadi)



Gambar 8 Komunitas Empat Simpul pada Graf Pemegang (Sumber: Arsip Pribadi)



Gambar 9 Komunitas Dua Simpul pada Graf Pemegang
(Sumber: Arsip Pribadi)

Dalam konteks penelitian ini, simpul utama, yaitu token itu sendiri, tidak dihitung sebagai bagian dari komunitas. Hal ini bermakna bahwa fokus utama adalah pada hubungan antar pemegang token, bukan pada hubungan mereka dengan token. Jumlah komunitas yang hampir sama dengan jumlah alamat pemegang menunjukkan bahwa sebagian besar pemegang token bertindak secara independen, tanpa adanya hubungan transaksional yang signifikan dengan pemegang lain. Ini berarti bahwa, walaupun ada banyak alamat pemegang, interaksi di antara mereka sangat terbatas.

Oleh karena itu, dalam analisis distribusi kepemilikan token, dapat dilakukan reinterpretasi jumlah pemegang token. Jika satu komunitas dianggap sebagai satu pemegang, ini menunjukkan bahwa distribusi kepemilikan token sebenarnya lebih tidak terdesentralisasi daripada yang diperkirakan sebelumnya. Dalam konteks ini, "pemegang" diartikan sebagai entitas independen yang keputusannya tidak dipengaruhi oleh pemegang lain. Dengan demikian, meskipun secara nominal terdapat 394 pemegang, secara fungsional jumlah pemegang aktual yang beroperasi secara independen adalah 360.

Pemahaman tentang desentralisasi dan independensi dalam kepemilikan token kripto merupakan aspek penting yang harus diperhatikan, terutama dalam analisis ekonomi kripto. Hasil klusterisasi yang menunjukkan adanya banyak komunitas pemegang token yang terpisah dan beroperasi secara mandiri mengindikasikan tingkat desentralisasi yang tinggi. Artinya, dalam konteks kepemilikan token, sebagian besar pemegang membuat keputusan investasi atau transaksi mereka tanpa pengaruh yang signifikan dari pemegang lain. Fenomena ini menegaskan bahwa tiap pemegang, dalam konteks pengambilan keputusan, bertindak secara individual dan otonom.

Dalam praktiknya, hal ini mencerminkan bahwa pasar token ini tidak didominasi oleh sekelompok pemegang tertentu yang dapat mempengaruhi pasar secara keseluruhan. Kondisi ini berbeda dengan situasi di mana beberapa pemegang besar memiliki pengaruh yang cukup signifikan terhadap harga dan likuiditas pasar. Oleh karena itu, pemahaman tentang tingkat desentralisasi ini sangat krusial dalam mengevaluasi stabilitas pasar dan potensi manipulasi harga.

IV. KESIMPULAN

Dalam menganalisis distribusi token di ekosistem blockchain Ethereum, metode yang efektif adalah melalui analisis graf dengan menerapkan algoritma klusterisasi Louvain. Hasil analisis ini dapat mengungkapkan bahwa mayoritas pemegang token dalam ekosistem ini beroperasi

secara independen dan membentuk sejumlah komunitas yang terisolasi atau memiliki keterkaitan yang minim satu sama lain. Temuan ini menggambarkan tingkat desentralisasi dalam kepemilikan token kripto di ekosistem Ethereum. Pendekatan analisis graf ini dapat diterapkan dalam berbagai ekosistem blockchain lainnya, bukan hanya terbatas pada Ethereum. Tingkat desentralisasi yang tinggi memiliki dampak positif terhadap stabilitas pasar, mengurangi potensi manipulasi harga yang dapat dilakukan oleh pemegang token besar. Oleh karena itu, pemahaman mendalam tentang distribusi token yang bersifat terdesentralisasi sangat penting dalam penilaian risiko dan potensi investasi di dalam ekosistem kripto. Prinsip dasar desentralisasi ini merupakan pondasi utama dalam teknologi blockchain dan memiliki peran sentral dalam mencapai tujuan keuangan yang lebih merata dan adil.

V. SARAN

Untuk penelitian selanjutnya, beberapa pendekatan berbasis graf dapat digunakan untuk menganalisis distribusi token kripto secara lebih akurat. Pertama, analisis jaringan *multilayer* dapat diaplikasikan untuk memahami berbagai jenis interaksi, seperti transaksi dan pertukaran, yang mempengaruhi distribusi token. Kedua, pendekatan analisis temporal akan berguna untuk menelusuri perubahan distribusi token seiring waktu, mengungkap dinamika komunitas. Ketiga, implementasi algoritma klusterisasi yang lebih canggih dapat membantu mengidentifikasi sub-komunitas dalam jaringan. Terakhir, integrasi data tambahan, seperti demografi pemegang token atau data pasar, akan memperkaya analisis dan memberikan wawasan yang lebih komprehensif tentang distribusi kepemilikan token. Pendekatan-pendekatan ini akan memberikan perspektif baru dan lebih mendalam tentang struktur pasar kripto.

VI. UCAPAN TERIMA KASIH

Penulis ingin mengungkapkan rasa syukur kepada Tuhan Yang Maha Esa atas berkat-Nya yang telah memungkinkan penyelesaian makalah berjudul "Analisis Graf untuk Mengungkap Distribusi Aktual Token di Ekosistem Blockchain Ethereum" tanpa kendala yang berarti. Penulis juga ingin mengucapkan terima kasih kepada Ibu Dr. Fariska Zakhralativa Ruskanda S.T., M.T., yang telah menjadi dosen pengajar dalam mata kuliah IF2120 Matematika Diskrit dan memberikan bimbingan selama satu semester. Penulis juga merasa berterima kasih kepada semua pihak yang telah memberikan dukungan, meskipun tidak dapat disebutkan satu per satu, yang turut membantu dalam menyelesaikan pembuatan makalah ini.

REFERENSI

- [1] R. Munir, "Graf (Bagian 1)," 2023. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf>. [Diakses: 8 Desember 2023].
- [2] "Ethereum," ConsenSys. [Online]. Tersedia: <https://consensys.io/knowledge-base/ethereum>. [Diakses: 8 Desember 2023].
- [3] "What Is ERC-20 and What Does It Mean for Ethereum?" Investopedia. [Online]. Tersedia:

<https://www.investopedia.com/news/what-erc20-and-what-does-it-mean-ethereum/>. [Diakses: 8 Desember 2023].

- [4] "What Is an Ethereum Address?" Etherscan. [Online]. Tersedia: <https://info.etherscan.com/what-is-an-ethereum-address/>. [Diakses: 8 Desember 2023].
- [5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, dan E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech., vol. 2008, no. 12, hal. P10008, 2008.
- [6] A. Lancichinetti dan S. Fortunato, "Community detection algorithms: A comparative analysis," Phys. Rev. E, vol. 80, no. 5, hal. 056117, Nov. 2009. doi: 10.1103/physreve.80.056117.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2023



Juan Alfred Widjaya
13522073