

Implementasi Teori Graf pada Bubble Map untuk Deteksi Bundle pada Token Solana

Malik Arsyafiandra Madani – 13525008

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13525008@std.stei.itb.ac.id, malikarsya123@gmail.com

Abstract—Token kripto di jaringan Solana, terutama meme-coin, sering diluncurkan dengan pola distribusi yang menyebarkan. Salah satu praktik yang umum adalah *bundling*, yaitu pembelian sebagian besar pasokan token oleh satu pihak melalui banyak dompet sekaligus saat peluncuran. Praktik ini menyembunyikan tingkat konsentrasi kepemilikan, sehingga investor sulit menilai risiko *rug pull*. Makalah ini membahas penerapan teori graf untuk mendeteksi bundle melalui representasi *bubble map* sebagai graf berarah. Dompet dimodelkan sebagai simpul dan relasi pendanaan antar-dompet sebagai sisi. Dengan algoritma pencarian komponen terhubung berbasis struktur *Union-Find* yang ditambah batasan temporal, kelompok dompet yang berasal dari sumber dana yang sama dan membeli token pada slot berdekatan dapat diidentifikasi sebagai bundle. Pendekatan ini diuji pada data *on-chain* dan menunjukkan bahwa graf mampu mengungkap konsentrasi kepemilikan yang tidak terlihat dari daftar *holder* biasa, dengan kompleksitas mendekati linear terhadap jumlah transaksi.

Index Terms—bundle, bubble map, graf berarah, komponen terhubung, Solana, Union-Find

I. PENDAHULUAN

Jaringan Solana dikenal karena biaya transaksi yang rendah dan waktu blok yang cepat. Sifat ini membuatnya menjadi tempat favorit untuk meluncurkan memecoin, yaitu token yang nilainya lebih didorong oleh spekulasi daripada utilitas. Setiap hari ada ribuan token baru yang muncul melalui platform peluncuran seperti Pump.fun. Sebagian besar di antaranya berumur sangat pendek dan tidak sedikit yang berakhir sebagai penipuan.

Salah satu pola penipuan yang sering ditemui adalah *bundle*. Pada pola ini, pembuat token (*deployer*) membeli sebagian besar pasokan token tepat pada saat peluncuran, tetapi pembelian itu dipecah ke banyak dompet yang berbeda. Tujuannya adalah menyamarkan konsentrasi kepemilikan. Jika dilihat sekilas, token tampak memiliki banyak *holder* sehingga terkesan terdistribusi secara wajar. Padahal seluruh dompet itu dikendalikan oleh satu pihak yang sama, dan suatu saat pasokan tersebut akan dijual serentak sehingga harga jatuh dan investor lain menanggung kerugian.

Untuk membantu analisis kepemilikan, muncul alat visualisasi seperti Bubblemaps. Alat ini menggambarkan setiap dompet sebagai gelembung yang ukurannya sebanding dengan jumlah token yang dipegang, lalu menghubungkan gelembung-gelembung yang saling bertransaksi dengan garis. Hasilnya adalah peta yang secara visual menarik, tetapi interpretasinya

masih bergantung pada penilaian manusia. Diperlukan metode formal agar deteksi bundle dapat dilakukan secara otomatis dan terukur.

Pendekatan yang hanya menjumlahkan kepemilikan pemegang teratas juga tidak memadai. Justru karena bundle dirancang untuk memecah pasokan ke banyak dompet kecil, kelemahan terbesar dari analisis daftar holder biasa adalah ketidakmampuannya melihat keterkaitan antar-dompet. Dibutuhkan cara yang menyoroti relasi, bukan sekadar nilai individu.

Relasi antar-dompet pada dasarnya adalah struktur graf: dompet adalah objek, dan transaksi pendanaan adalah hubungan antar objek. Karena itu, teori graf merupakan alat yang tepat untuk memodelkan dan menganalisis persoalan ini. Makalah ini mengusulkan pemodelan bubble map sebagai graf berarah berbobot, lalu menerapkan algoritma pencarian komponen terhubung yang dipadukan dengan batasan temporal untuk mengidentifikasi bundle. Kontribusi makalah ini adalah formalisasi definisi bundle dalam bahasa teori graf dan rancangan algoritma deteksinya beserta analisis kompleksitas.

Sisa makalah disusun sebagai berikut. Bagian II membahas landasan teori graf yang relevan. Bagian III menjelaskan pemodelan bubble map menjadi graf. Bagian IV menguraikan definisi formal bundle dan algoritma deteksinya. Bagian V membahas implementasi dan eksperimen. Bagian VI menyajikan analisis, dan Bagian VII memuat kesimpulan.

II. LANDASAN TEORI

A. Graf

Graf G adalah pasangan (V, E) dengan V himpunan tak kosong berisi simpul (*vertex*) dan E himpunan sisi (*edge*) yang menghubungkan pasangan simpul [1]. Jika setiap sisi memiliki arah, graf disebut graf berarah (*directed graph*). Sisi berarah dinyatakan sebagai pasangan terurut (u, v) yang berarti ada hubungan dari u ke v .

Pada persoalan ini, arah penting karena aliran dana memiliki sumber dan tujuan yang jelas. Sebuah dompet membiayai dompet lain, bukan sebaliknya. Selain berarah, graf yang dipakai juga berbobot, karena setiap sisi membawa informasi tambahan seperti jumlah SOL yang ditransfer dan waktu transaksi.

B. Representasi Graf

Sebuah graf dapat disimpan dengan matriks ketetanggaan (*adjacency matrix*) atau senarai ketetanggaan (*adjacency list*). Matriks ketetanggaan berukuran $n \times n$ sehingga memakan ruang $O(n^2)$. Cara ini boros bila graf jarang (*sparse*), yaitu ketika jumlah sisi jauh lebih kecil daripada n^2 .

Graf pendanaan pada token Solana termasuk graf jarang. Sebagian besar dompet hanya dibiayai oleh satu atau dua sumber, sehingga rata-rata derajatnya kecil. Karena itu, senarai ketetanggaan lebih hemat, dengan kebutuhan ruang $O(n + m)$ untuk n simpul dan m sisi. Representasi inilah yang dipakai pada implementasi.

C. Derajat dan Keterhubungan

Derajat-masuk (*in-degree*) sebuah simpul adalah jumlah sisi yang menuju simpul itu, sedangkan derajat-keluar (*out-degree*) adalah jumlah sisi yang keluar darinya. Pada graf pendanaan, simpul dengan derajat-keluar besar berpotensi menjadi sumber dana bersama yang mendistribusikan SOL ke banyak dompet.

Dua simpul dikatakan terhubung jika ada lintasan (*path*) di antara keduanya. Pada graf berarah, kita membedakan keterhubungan kuat dan keterhubungan lemah. Keterhubungan lemah (*weakly connected*) terpenuhi jika dua simpul terhubung ketika arah sisi diabaikan. Untuk deteksi bundle, keterhubungan lemah sudah cukup, karena yang dicari adalah keterkaitan pendanaan, bukan arah spesifiknya.

D. Komponen Terhubung

Komponen terhubung adalah himpunan bagian maksimal dari simpul yang setiap pasangannya saling terhubung [2]. Pada graf berarah, komponen terhubung lemah (*weakly connected component*, WCC) diperoleh dengan memperlakukan graf seolah tak berarah lalu mencari komponen-komponennya. Setiap WCC merepresentasikan satu kelompok dompet yang terkait secara pendanaan. Kelompok inilah yang menjadi kandidat bundle.

E. Struktur Union-Find

Struktur data *Union-Find* atau *Disjoint Set* dipakai untuk mengelola partisi himpunan menjadi kelompok-kelompok terpisah [3]. Struktur ini mendukung dua operasi utama: *FIND*, yang menentukan kelompok suatu elemen, dan *UNION*, yang menggabungkan dua kelompok. Dengan optimasi *path compression* dan *union by rank*, kedua operasi berjalan dalam waktu hampir konstan, yaitu $O(\alpha(n))$ dengan α adalah fungsi Ackermann invers yang nilainya sangat kecil untuk ukuran data praktis. Union-Find sangat cocok untuk mencari komponen terhubung secara bertahap saat sisi dibaca satu per satu.

F. Koefisien Pengelompokan dan Sentralitas

Koefisien pengelompokan (*clustering coefficient*) suatu simpul mengukur seberapa rapat tetangga-tetangganya saling terhubung [4]. Nilai yang tinggi menandakan kelompok yang padat, ciri khas dompet-dompet yang dibiayai dari sumber yang sama. Sentralitas derajat (*degree centrality*) mengukur pengaruh sebuah simpul berdasarkan banyaknya sisi yang

melekat padanya. Pada graf pendanaan, simpul dengan sentralitas tinggi biasanya merupakan dompet pembiaya utama.

G. Bundle dan Bubble Map

Pada konteks Solana, *bundle* merujuk pada sekelompok transaksi yang dieksekusi bersamaan dalam satu blok, sering memanfaatkan layanan seperti Jito agar pembelian terjadi pada slot yang sama. Secara ekonomi, bundle dipakai untuk membeli pasokan token serentak melalui banyak dompet sambil menghindari deteksi. *Bubble map* adalah representasi visual dari kepemilikan dan relasi transfer antar-dompet [8]. Makalah ini menjembatani keduanya: bubble map diperlakukan sebagai graf, lalu bundle dicari sebagai pola tertentu di dalam graf tersebut.

III. PEMODELAN BUBBLE MAP SEBAGAI GRAF

A. Sumber Data On-Chain

Data kepemilikan dan transaksi diperoleh dari jaringan Solana melalui pemanggilan RPC atau layanan terindeks seperti Helius [6], [7]. Untuk sebuah token, dua jenis data dikumpulkan: daftar pemegang token beserta jumlahnya, dan riwayat transfer SOL yang membiayai dompet-dompet tersebut sebelum mereka membeli token.

B. Definisi Simpul dan Sisi

Misalkan W adalah himpunan dompet yang pernah memegang token target. Setiap dompet menjadi satu simpul. Sisi pendanaan dibentuk dari transfer SOL: jika dompet u mengirim SOL ke dompet v sebelum v membeli token, maka dibentuk sisi berarah (u, v) . Graf pendanaan didefinisikan sebagai

$$G_f = (W, E_f), \quad E_f = \{(u, v) \mid u \text{ membiayai } v\}. \quad (1)$$

Setiap simpul v memiliki dua atribut penting: jumlah token yang dipegang $h(v)$, dan slot pembelian $s(v)$, yaitu nomor blok saat v pertama kali membeli token. Setiap sisi memiliki bobot berupa jumlah SOL yang ditransfer.

C. Aturan Pembentukan Sisi

Tidak semua transfer SOL layak menjadi sisi pendanaan. Transfer yang relevan adalah yang benar-benar membiayai pembelian token. Karena itu dipakai dua syarat. Pertama, transfer harus terjadi sebelum penerima membeli token, yaitu pada slot yang lebih kecil daripada $s(v)$. Kedua, jarak waktunya tidak boleh terlalu jauh; transfer dianggap relevan jika

$$0 \leq s(v) - t(u, v) \leq \Delta t, \quad (2)$$

dengan $t(u, v)$ slot saat transfer terjadi dan Δt jendela waktu yang ditentukan. Syarat ini menyingkirkan transfer lama yang kebetulan terjadi jauh sebelum peluncuran token.

Persoalan lain adalah dompet perantara (*pass-through*). Pelaku kadang menyalurkan dana melalui rantai beberapa dompet agar relasi tidak langsung terlihat. Selama setiap mata rantai memenuhi syarat di atas, dompet perantara tetap menjadi simpul, dan komponen terhubung lemah akan menyatukan seluruh rantai itu menjadi satu kelompok. Inilah keunggulan pendekatan graf: relasi tidak langsung pun tetap terungkap selama jalurnya terhubung.

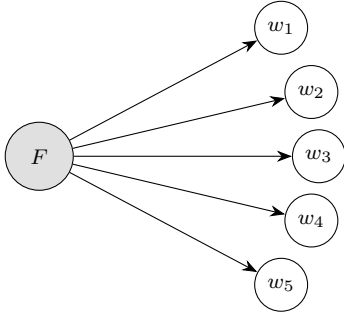


Fig. 1. Contoh graf pendanaan. Satu dompet pembiaya F membiayai lima dompet yang kemudian membeli token target.

D. Ilustrasi Graf Pendanaan

Gambar 1 menunjukkan contoh sederhana. Simpul F adalah dompet pembiaya yang mengirim SOL ke lima dompet (w_1 sampai w_5). Kelima dompet itu kemudian membeli token target. Karena semuanya berasal dari satu sumber dana dan terhubung lemah, mereka membentuk satu komponen yang patut dicurigai sebagai bundle.

IV. ALGORITMA DETEKSI BUNDLE

A. Definisi Formal Bundle

Misalkan $C \subseteq W$ adalah sebuah komponen terhubung lemah pada G_f . Komponen C disebut bundle jika memenuhi tiga syarat berikut:

- 1) Ukuran komponen melewati ambang batas: $|C| \geq \theta$.
- 2) Pembelian terjadi dalam rentang slot yang sempit:

$$\Delta s(C) = \max_{v \in C} s(v) - \min_{v \in C} s(v) \leq \tau. \quad (3)$$

- 3) Konsentrasi pasokan yang dikuasai melewati ambang:

$$\rho(C) = \frac{\sum_{v \in C} h(v)}{H_{\text{total}}} \geq \rho_{\min}, \quad (4)$$

dengan H_{total} adalah total pasokan token.

Syarat pertama menyaring kelompok yang terlalu kecil. Syarat kedua memastikan kelompok benar-benar membeli secara serentak, ciri utama bundle yang memakai eksekusi satu blok. Syarat ketiga memastikan kelompok itu menguasai pasokan dalam jumlah yang signifikan sehingga benar-benar berbahaya.

B. Langkah Algoritma

Algoritma deteksi terdiri atas empat tahap. Pertama, bangun graf pendanaan dari data transfer SOL. Kedua, cari semua komponen terhubung lemah menggunakan Union-Find. Ketiga, untuk setiap komponen, hitung rentang slot dan konsentrasi pasokan. Keempat, tandai komponen yang memenuhi ketiga syarat sebagai bundle. Rincian langkahnya disajikan pada Algoritma 1.

Algorithm 1 Deteksi Bundle pada Token Solana

Require: Himpunan transfer T , atribut h, s , ambang $\theta, \tau, \rho_{\min}$

Ensure: Himpunan komponen bundle B

```

1: Inisialisasi Union-Find untuk semua dompet di  $W$ 
2: for setiap transfer  $(u, v) \in T$  do
3:   if  $u$  mengirim SOL ke  $v$  sebelum  $v$  membeli then
4:     UNION( $u, v$ )
5:   end if
6: end for
7: Kelompokkan dompet berdasarkan FIND( $\cdot$ ) menjadi himpunan komponen  $C$ 
8:  $B \leftarrow \emptyset$ 
9: for setiap komponen  $C \in \mathcal{C}$  do
10:   $\Delta s \leftarrow \max_{v \in C} s(v) - \min_{v \in C} s(v)$ 
11:   $\rho \leftarrow (\sum_{v \in C} h(v)) / H_{\text{total}}$ 
12:  if  $|C| \geq \theta$  dan  $\Delta s \leq \tau$  dan  $\rho \geq \rho_{\min}$  then
13:     $B \leftarrow B \cup \{C\}$ 
14:  end if
15: end for
16: return  $B$ 

```

TABLE I
PENELUSURAN UNION-FIND PADA EMPAT TRANSFER

Operasi	w_1	w_2	w_3	w_4	w_5	w_6	w_7
awal	1	2	3	4	5	6	7
(w_1, w_2)	1	1	3	4	5	6	7
(w_1, w_3)	1	1	1	4	5	6	7
(w_3, w_4)	1	1	1	1	5	6	7
(w_5, w_6)	1	1	1	1	5	5	7

C. Analisis Kompleksitas

Misalkan $n = |W|$ jumlah dompet dan $m = |T|$ jumlah transfer. Membangun Union-Find membutuhkan $O(n)$. Memproses seluruh transfer membutuhkan m operasi UNION, masing-masing $O(\alpha(n))$, sehingga total $O(m\alpha(n))$. Pengelompokan dan evaluasi tiap komponen menelusuri setiap dompet satu kali, yaitu $O(n)$. Maka kompleksitas keseluruhan adalah

$$O(n + m\alpha(n)) \approx O(n + m), \quad (5)$$

yang praktis linear terhadap ukuran data. Sifat ini penting karena jumlah transaksi pada token populer bisa sangat besar.

D. Contoh Penelusuran

Untuk memperjelas cara kerja algoritma, perhatikan tujuh dompet w_1 sampai w_7 dengan urutan transfer berikut: (w_1, w_2) , (w_1, w_3) , (w_3, w_4) , (w_5, w_6) . Mula-mula setiap dompet menjadi kelompoknya sendiri. Tabel I menelusuri perubahan akar (*root*) setiap dompet setelah setiap operasi UNION.

Setelah semua transfer diproses, terbentuk tiga komponen: $\{w_1, w_2, w_3, w_4\}$ dengan akar w_1 , $\{w_5, w_6\}$ dengan akar w_5 , dan $\{w_7\}$ yang berdiri sendiri. Jika ambang ukuran $\theta = 4$, hanya komponen pertama yang lolos tahap ini dan diteruskan

ke pemeriksaan temporal serta konsentrasi. Contoh ini memperlihatkan bagaimana banyak dompet yang tampak terpisah pada daftar holder sebenarnya menyatu menjadi satu kelompok begitu relasi pendanaan dipetakan.

E. Alternatif Pencarian Komponen

Selain Union-Find, komponen terhubung juga dapat dicari dengan penelusuran melebar (*Breadth-First Search*, BFS) atau penelusuran mendalam (*Depth-First Search*, DFS). Pada pendekatan ini, graf disimpan sebagai senarai ketetanggaan tak berarah, lalu setiap simpul yang belum dikunjungi menjadi titik awal satu penelusuran. Semua simpul yang terjangkau dari titik itu membentuk satu komponen. Kompleksitasnya $O(n + m)$, sebanding dengan Union-Find. Union-Find tetap dipilih karena dapat memproses sisi secara bertahap saat data dibaca dari API, tanpa perlu menyimpan seluruh graf lebih dulu.

F. Pemodelan Bipartit Funder–Buyer

Persoalan ini juga dapat dipandang sebagai graf bipartit. Himpunan simpul dibagi menjadi dua: dompet pembiaya dan dompet pembeli. Sisi hanya menghubungkan pembiaya dengan pembeli yang didanainya. Pada graf bipartit, dua pembeli yang terhubung ke pembiaya yang sama berbagi tetangga. Bundle kemudian dapat dilihat sebagai subgraf bipartit padat, yaitu sekelompok pembeli yang seluruhnya terhubung ke satu pembiaya yang sama.

Cara pandang ini berguna karena menyoroti peran pembiaya sebagai pusat. Satu pembiaya dengan derajat tinggi yang mendanai banyak pembeli pada waktu berdekatan adalah pola bundle yang paling jelas. Dengan menelusuri tetangga dari setiap pembiaya berderajat tinggi, kandidat bundle dapat ditemukan tanpa membentuk seluruh komponen terlebih dahulu.

G. Keterkaitan dengan Deteksi Komunitas

Pada tingkat yang lebih umum, deteksi bundle merupakan bentuk khusus dari deteksi komunitas (*community detection*) pada graf. Komunitas adalah kelompok simpul yang relasinya jauh lebih rapat di dalam kelompok daripada ke luar kelompok. Bundle persis seperti itu: dompetnya saling terikat melalui pendanaan bersama, tetapi nyaris tidak punya relasi ke holder organik. Bedanya, deteksi bundle menambahkan batasan temporal dan konsentrasi yang khas pada konteks token, sehingga tidak setiap komunitas dianggap berbahaya. Sudut pandang ini membuka peluang memakai metode deteksi komunitas yang lebih canggih bila pola penayaman semakin rumit. Definisi berbasis komponen terhubung lemah dapat menghasilkan komponen yang terlalu besar bila ada dompet penghubung yang tidak relevan. Untuk itu, dapat ditambahkan analisis sumber dana bersama (*common funding ancestor*). Sebuah simpul a disebut leluhur pendanaan bagi v dalam k hop jika ada lintasan berarah dari a ke v pada G_f yang panjangnya paling banyak k . Komponen dianggap bundle yang kuat jika seluruh anggotanya berbagi setidaknya satu leluhur bersama a dengan

$$\forall v \in C, \quad d_{G_f}(a, v) \leq k, \quad (6)$$

```
# gabung dua dompet ke komponen yang sama
def union(parent, rank, a, b):
    ra, rb = find(parent, a), find(parent, b)
    if ra == rb: return
    if rank[ra] < rank[rb]: ra, rb = rb, ra
    parent[rb] = ra
    if rank[ra] == rank[rb]: rank[ra] += 1

# tandai komponen sebagai bundle
def is_bundle(comp, slot, hold, total, th, tau, rho):
    spread = max(slot[v] for v in comp) \
        - min(slot[v] for v in comp)
    conc = sum(hold[v] for v in comp) / total
    return len(comp) >= th and spread <= tau \
        and conc >= rho
```

Fig. 2. Cuplikan implementasi Union-Find dan penandaan bundle.

dengan $d_{G_f}(a, v)$ panjang lintasan terpendek dari a ke v . Leluher bersama ini dapat dicari dengan BFS dari kandidat simpul pembiaya, yaitu simpul berderajat-keluar tinggi.

Koefisien pengelompokan juga dapat dipakai sebagai sinyal tambahan. Untuk simpul v berderajat $\deg(v)$ dengan $L(v)$ banyaknya sisi yang menghubungkan antar-tetangganya, koefisiennya adalah

$$CC(v) = \frac{2 L(v)}{\deg(v) (\deg(v) - 1)}. \quad (7)$$

Kelompok bundle cenderung memiliki koefisien tinggi karena dompet-dompetnya saling terkait melalui pembiaya yang sama, sedangkan holder organik nyaris tidak membentuk segitiga semacam itu.

V. IMPLEMENTASI DAN EKSPERIMEN

A. Prototipe

Prototipe ditulis dalam Python. Pengambilan data *on-chain* memakai API Helius, sedangkan pembentukan dan analisis graf memakai pustaka NetworkX. Komponen terhubung dihitung dengan implementasi Union-Find sederhana agar pemrosesan transfer bisa dilakukan secara bertahap. Cuplikan inti algoritma ditunjukkan di bawah ini.

B. Pengaturan Parameter

Ketiga ambang θ , τ , dan $\rho_{\min}$ menentukan seberapa ketat algoritma menandai bundle. Ambang ukuran θ menyaring kelompok kecil; nilai yang terlalu rendah membuat dua atau tiga dompet biasa ikut tertandai, sedangkan nilai yang terlalu tinggi melewatkan bundle berukuran sedang. Nilai $\theta = 4$ dipilih sebagai titik tengah yang masuk akal untuk token tahap awal.

Ambang temporal τ mengatur seberapa rapat pembelian harus terjadi. Bundle yang memakai eksekusi satu blok lewat Jito jatuh pada slot yang sama. Namun klaster yang terkoordinasi pada peluncuran sering tersebar pada rentang beberapa puluh slot, sehingga τ perlu disetel selebar window peluncuran, bukan satu slot. Memperbesar τ menaikkan sensitivitas tetapi juga menaikkan risiko menyatukan pembeli yang sebenarnya tidak terkait dan kebetulan membeli berdekatan. Ambang konsentrasi $\rho_{\min}$ memastikan hanya kelompok yang menguasai pasokan signifikan yang dianggap berbahaya; kelompok

TABLE II
KOMPONEN UTAMA HASIL DETEKSI PADA TOKEN UJI (200 PEMBELI
PELUNCURAN, 174 KOMPONEN)

Komponen	Ukuran	Spread (slot)	ρ	Status
1	1 dompet	0	0,489	konsentrasi tunggal
2	24 dompet	36	0,063	bundle
3	1 dompet	0	0,005	wajar

kecil yang terhubung tetapi hanya memegang sedikit token tidak perlu dirisaukan.

Karena pemilihan ambang bergantung pada karakter token, nilai-nilai ini sebaiknya tidak dipatok kaku. Pendekatan yang lebih baik adalah membuatnya adaptif terhadap distribusi pasokan token yang sedang diperiksa.

C. Studi Kasus

Pengujian dilakukan pada satu token nyata di Solana dengan `mint HYbGxqQp...Lo2PD`. Data diambil melalui API Helius dengan menelusuri transaksi token sampai ke blok peluncuran, lalu mengambil pembeli pada window peluncuran. Dari pembeli tersebut, 200 dompet dianalisis dan terbentuk 174 komponen terhubung. Parameter yang dipakai adalah $\theta = 4$, $\tau = 40$ slot, dan $\rho_{\min} = 0,05$. Nilai τ disetel selebar rentang aktivitas peluncuran, bukan satu slot, agar klaster yang membeli pada window yang sama tetap tertangkap.

Hasilnya memperlihatkan dua pola yang menonjol. Komponen terbesar hanya berisi satu dompet yang menguasai 48,9% pasokan. Pola ini bukan bundle, melainkan konsentrasi oleh satu pihak tunggal, dan memang tidak lolos syarat ukuran θ . Komponen terbesar berikutnya berisi 24 dompet yang terhubung melalui pendanaan bersama dan aktif pada window peluncuran dengan rentang 36 slot. Kelompok ini menguasai 6,3% pasokan dan memenuhi ketiga syarat, sehingga ditandai sebagai bundle. Sisa komponen umumnya berisi satu dompet dengan porsi sangat kecil. Tabel II merangkum komponen-komponen utama.

D. Visualisasi Hasil

Gambar 3 menggambarkan dua pola pada token uji. Di sisi kiri, klaster 24 dompet saling terhubung melalui pendanaan bersama dan membentuk subgraf padat. Di sisi kanan, satu dompet dominan berdiri sendiri tanpa relasi pendanaan ke dompet lain. Klaster padat itulah yang ditandai sebagai bundle, sedangkan dompet dominan ditangani terpisah sebagai konsentrasi tunggal.

VI. ANALISIS DAN PEMBAHASAN

Hasil menunjukkan bahwa pemodelan graf mampu mengubah persoalan deteksi bundle yang semula bergantung pada penilaian visual menjadi pemeriksaan yang terukur. Konsentrasi kepemilikan yang tersembunyi di balik banyak dompet menjadi terlihat begitu relasi pendanaan dipetakan sebagai sisi graf.

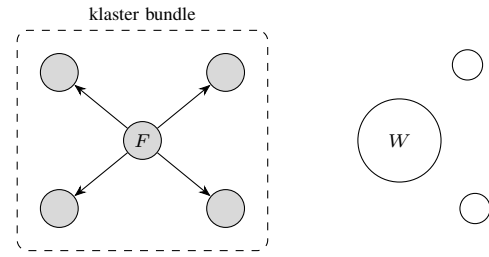


Fig. 3. Dua pola pada token uji. Kiri: klaster bundle (kotak putus-putus) yang dompet-dompetnya terhubung lewat pendanaan bersama. Kanan: dompet dominan tunggal W yang berdiri sendiri.

A. Perbandingan dengan Metode Naif

Metode paling sederhana untuk menilai konsentrasi adalah melihat daftar pemegang teratas. Pada token uji, cara ini langsung menyoroti dompet yang menguasai 48,9% pasokan, tetapi melewatkan 24 dompet pada klaster bundle. Sebabnya, masing-masing dompet itu hanya memegang porsi kecil sehingga tampak sebagai pemegang yang terpisah, dan daftar holder memperlakukannya sebagai 24 entitas berbeda. Graf pendanaan menyatukan ke-24 dompet itu menjadi satu kelompok terkoordinasi yang, bila digabung, menguasai 6,3% pasokan dari aktivitas yang seragam pada peluncuran. Perbedaan ini menegaskan bahwa nilai tambah graf terletak pada kemampuannya menyatukan dompet yang terkait, bukan sekadar menilai dompet satu per satu.

B. Kelebihan dan Keterbatasan

Kompleksitasnya hampir linear, sehingga sanggup memproses token dengan ribuan transaksi. Hasilnya juga mudah ditafsirkan, karena setiap bundle yang ditemukan dapat ditelusuri kembali ke sumber dananya. Metode ini tidak bergantung pada pelatihan model, sehingga bisa langsung dipakai pada token baru tanpa data historis.

Keterbatasan utamanya adalah kemungkinan *false positive*. Dompet milik bursa terpusat (*centralized exchange*) sering menjadi simpul berderajat sangat tinggi karena membiayai banyak pengguna yang tidak saling terkait. Tanpa penanganan, simpul semacam ini bisa menyatukan dompet-dompet yang sebenarnya independen menjadi satu komponen raksasa. Solusinya adalah menyusun daftar putih (*whitelist*) alamat bursa dan layanan publik, lalu mengeluarkannya dari graf sebelum pencarian komponen.

Tantangan lain adalah upaya penyamaran. Pelaku dapat menyebar pendanaan melalui banyak hop atau memakai jeda waktu agar slot pembelian tidak terlalu rapat. Penyempurnaan berbasis sumber dana bersama dan pelanggaran batas temporal yang adaptif dapat membantu, meski selalu ada keseimbangan antara sensitivitas dan tingkat kesalahan.

Hasil token uji juga memperlihatkan dua batasan yang perlu disadari. Pertama, metode ini secara sengaja membedakan bundle dari konsentrasi tunggal: dompet yang menguasai 48,9% pasokan tidak ditandai sebagai bundle karena hanya satu simpul, padahal risikonya besar. Karena itu, pemeriksaan konsentrasi per dompet sebaiknya dijalankan berdampingan

dengan deteksi bundle. Kedua, deteksi bergantung pada adanya sisi pendanaan; bila pembiayaan dilakukan lewat layanan pihak ketiga yang tidak meninggalkan transfer langsung antar dompet, klaster bisa terpecah dan luput. Menambahkan sisi berdasarkan kesamaan slot pembelian dapat melengkapi sinyal pendanaan untuk kasus seperti ini.

VII. KESIMPULAN

Makalah ini menunjukkan bahwa teori graf dapat dipakai untuk mendeteksi bundle pada token Solana secara formal. Dengan memodelkan bubble map sebagai graf pendanaan berarah, bundle dapat didefinisikan sebagai komponen terhubung lemah yang memenuhi syarat ukuran, kedekatan temporal, dan konsentrasi pasokan. Algoritma yang diusulkan memakai struktur Union-Find dan berjalan dalam waktu hampir linear terhadap jumlah transaksi. Pengujian pada token contoh memperlihatkan bahwa metode ini mampu memisahkan token yang ter-bundle dari yang terdistribusi wajar. Pengembangan selanjutnya dapat mengarah pada penanganan penyamaran multi-hop dan integrasi sinyal koefisien pengelompokan untuk menekan kesalahan deteksi.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, kepada dosen mata kuliah IF1220 Matematika Diskrit atas ilmu yang diberikan, serta kepada keluarga dan teman-teman yang mendukung penyusunan makalah ini.

REFERENCES

- [1] R. Munir, *Matematika Diskrit*, Edisi Revisi Kelima. Bandung: Penerbit Informatika, 2016.
- [2] K. H. Rosen, *Discrete Mathematics and Its Applications*, 8th ed. New York: McGraw-Hill, 2019.
- [3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge: MIT Press, 2009.
- [4] M. E. J. Newman, *Networks: An Introduction*. Oxford: Oxford University Press, 2010.
- [5] F. Reid and M. Harrigan, "An Analysis of Anonymity in the Bitcoin System," in *Security and Privacy in Social Networks*. New York: Springer, 2013, pp. 197–223.
- [6] Solana Foundation, "Solana Documentation," <https://docs.solana.com>, diakses 17 Juni 2026.
- [7] Helius, "Helius API Documentation," <https://docs.helius.dev>, diakses 17 Juni 2026.
- [8] Bubblemaps, "Bubblemaps: Visualize Blockchain Data," <https://bubblemaps.io>, diakses 17 Juni 2026.

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, 19 Juni 2026



Malik Arsyafiandra Madani – 13525008