

Analisis Performa Algoritma Keputusan Berbasis Memori pada Berbagai Topologi Graf: Simulasi Iterated Prisoner's Dilemma

Moch Naufal Zaki Basara - 13525095

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: naufaly57@gmail.com , 13525095@std.stei.itb.ac.id

Abstract—Penelitian ini menganalisis pengaruh topologi graf terhadap performa empat algoritma keputusan berbasis memori — *Always Defect* (AllD), *Always Cooperate* (AllC), *Tit-for-Tat* (TfT), dan *Pavlov* — dalam simulasi *Iterated Prisoner's Dilemma*. Simulasi dilakukan pada lima topologi graf: *Complete Graph*, *Random Graph* (Erdős-Rényi), *Scale-Free Graph* (Barabási-Albert), *Zachary's Karate Club*, dan *ego-Facebook*, masing-masing dieksekusi sebanyak lima kali. Hasil menunjukkan bahwa AllD secara konsisten mendominasi seluruh topologi akibat tidak adanya mekanisme evolusi strategi, namun derajat keunggulannya bervariasi signifikan antar topologi, dengan gap terkecil terjadi pada *Scale-Free* dan *Karate Club*. Pada dua kasus eksperimen, algoritma TfT bahkan berhasil mengungguli AllD, khususnya pada topologi dengan jumlah node terbatas dan variabilitas struktural tinggi. Temuan ini menunjukkan bahwa topologi graf mempengaruhi performa algoritma keputusan, meskipun tidak mengubah algoritma yang dominan secara keseluruhan.

Keywords—*Iterated Prisoner's Dilemma*, *topologi graf*, *algoritma keputusan*, *teori graf*, *clustering coefficient*

I. PENDAHULUAN

Sistem multi-agen merupakan sistem yang terdiri atas banyak entitas otonom yang berinteraksi dan mengambil keputusan secara independen. Performa keseluruhan sistem semacam ini dipengaruhi oleh dua faktor utama seperti algoritma keputusan yang digunakan oleh masing-masing agen dan struktur interaksi yang menghubungkan agen-agen tersebut. Salah satu model matematis yang umum digunakan untuk mempelajari interaksi strategis antar agen adalah *Iterated Prisoner's Dilemma* (IPD), sementara struktur interaksi antar agen dapat direpresentasikan menggunakan model graf, di mana setiap agen merupakan simpul dan setiap interaksi yang mungkin terjadi direpresentasikan sebagai sisi.

Penelitian Axelrod (1980) mempelopori studi mengenai performa algoritma keputusan pada IPD melalui turnamen round-robin, di mana ditemukan bahwa algoritma *Tit-for-Tat* (TfT) berhasil mengungguli algoritma-algoritma lain. Namun, turnamen tersebut tidak mempertimbangkan struktur jaringan sama sekali. Penelitian Nowak & May (1992) kemudian

memperkenalkan struktur spasial ke dalam simulasi IPD, namun fokus utama penelitian tersebut adalah efek evolusi strategi terhadap pembentukan kluster kooperatif, bukan perbandingan sistematis antar berbagai jenis topologi graf. Sejauh penelusuran penulis, belum banyak penelitian yang secara eksplisit membandingkan pengaruh berbagai bentuk topologi graf, baik sintetis maupun nyata, terhadap performa algoritma keputusan berbasis memori tanpa melibatkan evolusi strategi sebagai variabel pembaur.

Berdasarkan gap tersebut, penelitian ini bertujuan untuk menganalisis performa empat algoritma keputusan berbasis memori — *Always Defect* (AllD), *Always Cooperate* (AllC), *Tit-for-Tat*, dan *Pavlov* — pada lima topologi graf yang berbeda meliputi *Complete Graph*, *Random Graph* (Erdős-Rényi), *Scale-Free Graph* (Barabási-Albert), serta dua dataset jaringan nyata yaitu *Zachary's Karate Club* dan *ego-Facebook*. Penelitian ini secara spesifik berupaya menjawab sebuah pertanyaan, “apakah topologi graf dapat mempengaruhi konvergensi dan performa algoritma keputusan berbasis memori dalam simulasi *Iterated Prisoner's Dilemma*?”

Kontribusi penelitian ini meliputi: (1) analisis komparatif performa algoritma pada lima topologi graf yang berbeda karakteristik strukturalnya, (2) penggunaan dataset jaringan nyata berskala besar (*ego-Facebook*, 4.039 simpul) sebagai validasi terhadap graf sintetis, dan (3) analisis kuantitatif hubungan antara properti struktural graf — yaitu rata-rata derajat dan *clustering coefficient* — dengan performa relatif masing-masing algoritma.

Sistematika penulisan makalah ini adalah sebagai berikut. Bab II memaparkan dasar teori mengenai graf dan *Iterated Prisoner's Dilemma*. Bab III menjelaskan model dan metodologi penelitian. Bab IV menguraikan implementasi simulasi. Bab V menyajikan hasil simulasi beserta analisisnya. Bab VI memuat kesimpulan, keterbatasan, dan arah penelitian lanjutan.

II. DASAR TEORI

A. Teori Graf

1. Definisi Dasar

Graf didefinisikan sebagai $G = (V, E)$, di mana V adalah himpunan simpul (*node*) dan E adalah himpunan sisi (*edge*) yang menghubungkan pasangan simpul. Setiap sisi $(u, v) \in E$ merepresentasikan hubungan langsung antara simpul u dan simpul v .

Derajat (*degree*) suatu simpul v , dinotasikan $\deg(v)$, adalah jumlah sisi yang terhubung langsung dengan simpul tersebut — atau secara sederhana, jumlah tetangga langsung yang dimiliki simpul v .

Sebuah graf dikatakan terhubung (*connected*) apabila terdapat lintasan (*path*) antara setiap pasangan simpul di dalamnya. Graf lengkap (*complete graph*), dinotasikan K_n , adalah graf di mana setiap simpul terhubung langsung dengan seluruh simpul lainnya, sehingga $\deg(v) = n-1$ untuk setiap simpul v .

2. Degree Distribution

Degree distribution, dinotasikan $P(k)$, adalah probabilitas sebuah simpul yang dipilih secara acak memiliki derajat tepat k . Karakteristik degree distribution membedakan jenis-jenis graf secara matematis.

Model Erdős–Rényi (Random Graph) membentuk graf dengan menghubungkan setiap pasangan simpul secara independen dengan probabilitas p . Degree distribution yang dihasilkan mengikuti distribusi binomial, yang untuk n besar mendekati distribusi Poisson. Karakteristik utama model ini adalah derajat antar simpul relatif seragam — sebagian besar simpul memiliki derajat mendekati nilai rata-rata $E[\deg] = p(n-1)$, tanpa adanya simpul dengan derajat ekstrem.

Model Barabási–Albert (Scale-Free) dibentuk melalui mekanisme preferential attachment, di mana simpul baru yang ditambahkan ke dalam graf lebih cenderung terhubung dengan simpul yang sudah memiliki banyak koneksi. Mekanisme ini menghasilkan degree distribution yang mengikuti hukum pangkat (*power law*), $P(k) \sim k^{-(\gamma)}$, dengan karakteristik mayoritas simpul memiliki derajat rendah sementara segelintir simpul disebut *hub* memiliki derajat yang sangat tinggi. Struktur ini bersifat *robust-yet-fragile* — tahan terhadap kegagalan acak pada simpul berderajat rendah, namun rentan apabila *hub* mengalami kegagalan.

3. Clustering Coefficient

Clustering coefficient mengukur sejauh mana tetangga-tetangga suatu simpul juga saling terhubung satu sama lain. Untuk simpul v dengan k tetangga, clustering coefficient didefinisikan sebagai:

$$C(v) = (\text{jumlah sisi yang ada di antara tetangga } v) / (\text{jumlah sisi maksimum yang mungkin di antara tetangga } v)$$
 di mana jumlah sisi maksimum di antara k tetangga adalah $k(k-1)/2$. Nilai average clustering coefficient suatu graf adalah rata-rata $C(v)$ untuk seluruh simpul $v \in V$. Graf dengan clustering coefficient tinggi mengindikasikan adanya struktur komunitas,

di mana tetangga suatu simpul cenderung saling mengenal satu sama lain.

4. Lima Topologi yang Digunakan

Penelitian ini menggunakan lima topologi graf yang terbagi menjadi dua kategori. Tiga topologi pertama merupakan graf sintetis yang dibangkitkan secara komputasional: Complete Graph K_{100} , Random Graph Erdős–Rényi $G(100, 0,08)$, dan Scale-Free Graph Barabási–Albert dengan parameter $m=3$.

Dua topologi lainnya merupakan dataset jaringan nyata. Zachary's Karate Club adalah jaringan pertemanan 34 anggota klub karate yang dikumpulkan oleh Zachary (1977), tersedia sebagai dataset bawaan pada library NetworkX. Ego-Facebook adalah jaringan pertemanan Facebook anonim yang dikumpulkan oleh Stanford SNAP (Leskovec & Krevl, 2014), terdiri atas 4.039 simpul dan 88.234 sisi.

B. Iterated Prisoner's Dilemma

1. Payoff Matrix

Prisoner's Dilemma adalah model permainan dua pemain dengan dua pilihan: Cooperate (C) atau Defect (D). Hasil dari setiap kombinasi pilihan ditentukan oleh payoff matrix sebagai berikut:

	Lawan C	Lawan D
Pilih C	(3, 3)	(0, 5)
Pilih D	(5, 0)	(1, 1)

2. Defect sebagai Dominant Strategy pada One-Shot Game

Pada permainan satu kali (*one-shot game*), *Defect* merupakan dominant strategy karena memberikan hasil terbaik terlepas dari pilihan lawan. Hal ini dapat dibuktikan melalui dua skenario:

Skenario 1 — Lawan memilih C:

Kita pilih C $\rightarrow$ 3 poin

Kita pilih D $\rightarrow$ 5 poin (lebih baik)

Skenario 2 — Lawan memilih D:

Kita pilih C $\rightarrow$ 0 poin

Kita pilih D $\rightarrow$ 1 poin (lebih baik)

Karena Defect selalu menghasilkan poin lebih tinggi apapun pilihan lawan, setiap pemain rasional akan memilih Defect.

Hal ini memunculkan paradoks yang menjadi inti dilema permainan ini. Jika kedua pemain memilih *Cooperate*, masing-masing mendapat 3 poin, tetapi jika kedua pemain memilih *Defect* sebagai dominant strategy, masing-masing hanya akan mendapat 1 poin. Keputusan rasional secara individu justru menghasilkan kondisi yang lebih buruk bagi kedua pemain dibandingkan apabila mereka bekerja sama.

3. Iterasi Membuka Ruang Kooperasi

Pada one-shot game, *Defect* memang mendominasi dan tetap menang secara head-to-head ketika melawan *TfT*. Namun pada iterated game, meskipun *AllD* menang head-to-head melawan *TfT*, *AllD* tetap merugi secara total karena setelah ronde pertama *TfT* akan membalas dengan *Defect* dan berlanjut hingga ronde terakhir, sehingga *AllD* hanya mendapat payoff rendah (1,1) secara terus-menerus. Sebaliknya, jika kedua pemain memilih *TfT*, keduanya akan tetap berada pada kondisi *Cooperate* (3,3) sepanjang permainan, menghasilkan total payoff yang jauh lebih tinggi dibandingkan skenario *AllD* melawan *TfT*. Hal ini menunjukkan bahwa pada permainan yang diulang, strategi yang mampu membalas pengkhianatan membuka peluang akumulasi payoff yang lebih besar melalui kooperasi jangka panjang, sehingga *AllD* tidak lagi otomatis menjadi pilihan terbaik.

4. Empat Algoritma Keputusan

Penelitian ini menyimulasikan empat algoritma keputusan dengan karakteristik memori yang berbeda:

- Always Defect (*AllD*) — algoritma memori-0 yang selalu memilih *Defect* tanpa memperhatikan riwayat interaksi.
- Always Cooperate (*AllC*) — algoritma memori-0 yang selalu memilih *Cooperate* tanpa memperhatikan riwayat interaksi.
- Tit-for-Tat (*TfT*) — algoritma memori-1 yang memilih *Cooperate* pada ronde pertama, kemudian meniru pilihan lawan pada ronde sebelumnya.
- Pavlov (Win-Stay Lose-Shift) — algoritma memori-1 yang mengulang pilihan sebelumnya apabila payoff yang diterima ≥ 3 , dan mengganti pilihan apabila payoff < 3 .

III. MODEL DAN METODOLOGI

A. Formalisasi Model

Simulasi ini direpresentasikan dengan model graf tak-berarah $G = (V, E)$, dengan V adalah himpunan node yang merepresentasikan agen dan E adalah himpunan sisi yang merepresentasikan hubungan interaksi antar agen. Setiap node $v \in V$ memiliki dua atribut utama:

- Algoritma keputusan $a_v \in \{AllD, AllC, TfT, Pavlov\}$ yang ditetapkan secara acak pada awal simulasi dengan peluang seragam 0,25.

- Skor kumulatif $p_v \in \mathbb{R}$ yang diinisialisasi bernilai nol.

Pada setiap ronde simulasi, setiap sisi $(u, v) \in E$ merepresentasikan satu pertandingan Prisoner's Dilemma antara node u dan node v menggunakan algoritma masing-masing. Algoritma yang ditetapkan pada setiap node bersifat tetap sepanjang simulasi dan tidak berubah antar ronde. Skor kumulatif setiap node akan digunakan sebagai metrik performa algoritma yang digunakannya.

B. Pseudocode Simulasi

```

INPUT : Graf  $G = (V, E)$ , jumlah
ronde  $R$ 
OUTPUT: Skor kumulatif  $p_v$  untuk
setiap  $v \in V$ 

INISIALISASI:
1. Untuk setiap  $v \in V$ :
     $a_v \leftarrow \text{random}(\{AllD, AllC, TfT, Pavlov\})$ 
     $p_v \leftarrow 0$ 
     $last_v \leftarrow \text{None}$ 
SIMULASI:
2. Untuk setiap ronde  $r = 1, 2, \dots, R$ :
     $new\_last \leftarrow \{\}$ 

    Untuk setiap sisi  $(u, v) \in E$ :
         $m_u \leftarrow \text{get\_move}(a_u, last_u, last_v)$ 
         $m_v \leftarrow \text{get\_move}(a_v, last_v, last_u)$ 

         $(pay_u, pay_v) \leftarrow \text{PAYOFF}[(m_u, m_v)]$ 
         $p_u \leftarrow p_u + pay_u$ 
         $p_v \leftarrow p_v + pay_v$ 

         $new\_last[u] \leftarrow (m_u, pay_u)$ 
         $new\_last[v] \leftarrow (m_v, pay_v)$ 

     $last \leftarrow new\_last$ 

3. Return  $\{p_v \mid v \in V\}$ 

```

Fungsi `get_move`:

```

get_move(strategi, hasil_sendiri,
hasil_lawan):
    if strategi = AllD:
        return D
    if strategi = AllC:
        return C

```

```

    if strategi = Tft:
        if hasil_lawan = None: //
ronde pertama
            return C
        return move(hasil_lawan) //
tiru lawan
    if strategi = Pavlov:
        if hasil_sendiri = None: //
ronde pertama
            return C
        if payoff(hasil_sendiri) ≥
3:
            return
move(hasil_sendiri) // Win-Stay
        else:
            return
lawan(move(hasil_sendiri)) //
Lose-Shift

```

C. Parameter Eksperimen

Eksperimen dilakukan di 5 topologi graf, parameternya dapat dilihat pada Tabel 1.

Tabel 1. Parameter Eksperimen

Parameter	Nilai
Jumlah Node graf sintetis (N)	100
Probabilitas koneksi Erdős–Rényi (p)	0,08
Parameter attachment Barabási–Albert (m)	3
Jumlah ronde — graf sintetis (R)	100
Jumlah ronde — ego-Facebook (R)	20
Jumlah pengulangan — graf sintetis	10
Jumlah pengulangan — ego-Facebook	3
Random seed	42

Alasan pemilihan parameter:

- $p = 0.08$ dipilih agar expected degree ≈ 8 , sebanding dengan avg_degree Scale-Free (5.82) untuk perbandingan yang adil.

- $m = 3$ adalah nilai standar pada literatur Barabási–Albert untuk jaringan skala menengah.
- Seed 42 digunakan untuk reproduisibilitas — graf sintetis yang dihasilkan selalu identik tiap run.
- Facebook dibatasi 20 ronde dan 3 pengulangan karena keterbatasan komputasi (jika melakukan 100 ronde maka: $88.234 \text{ sisi} \times 100 \text{ ronde} = 8,8 \text{ juta interaksi per run}$).

D. Properti Struktural Graf

Sebelum mulai simulasi, properti setiap topologi diperlukan untuk keperluan analisis korelasi. Hasil pengukuran tercantum pada Tabel 2.

Tabel 2. Properti Struktural Graf

Topologi	Node	Edge	Avg degree	Avg Clustering
Complete	100	4.950	99,00	1
Random (ER)	100	371	7,42	0,065
Scale-free (BA)	100	291	5,82	0,187
Karate Club	34	78	4,59	0,571
ego-Facebook	4.039	88.234	43,69	0,606

E. Metrik Evaluasi

Tiga metrik digunakan untuk mengevaluasi performa algoritma:

- Rata-rata payoff
Skor kumulatif rerata node yang menggunakan algoritma a pada topologi G:

$$\bar{p}(a, G) = (1/|V_a|) \times \sum p_v \text{ untuk semua } v \in V_a$$

di mana $V_a \subseteq V$ adalah himpunan node dengan algoritma a.

- Standar deviasi
Mengukur konsistensi hasil setiap pengulangan. Std rendah berarti hasilnya stabil dan dapat diandalkan.
- Korelasi struktural
Analisis hubungan antara avg_degree dan avg_clustering dengan performa relatif antar algoritma per topologi.

IV. IMPLEMENTASI

A. Bahasa dan Library

Simulasi diimplementasikan menggunakan Python 3.12 dengan bantuan dua library utama: NetworkX 3.6.1 untuk pembangkitan dan analisis struktur graf, serta Matplotlib 3.11 untuk visualisasi hasil.

B. Pembangkitan Topologi

Tiga topologi sintetis dibangkitkan menggunakan fungsi bawaan NetworkX, sementara dua topologi nyata dimuat dari sumber eksternal:

```
N = 100 # jumlah node untuk graf
sintetis

G_complete = nx.complete_graph(N)
G_random = nx.erdos_renyi_graph(N,
0.08, seed=42)
G_scalefree = nx.barabasi_albert_graph(N,
3, seed=42)
G_karate = nx.karate_club_graph()
G_facebook =
nx.read_edgelist('facebook_combined.txt',
nodetype=int)
```

Parameter seed=42 digunakan pada graf sintetis untuk menjamin reproduksibilitas struktur graf antar eksekusi program.

C. Payoff Matrix

Hasil setiap interaksi antar dua node direpresentasikan menggunakan struktur dictionary yang memetakan kombinasi pilihan ke pasangan payoff:

```
PAYOFF = {
    ('C', 'C'): (3, 3),
    ('C', 'D'): (0, 5),
    ('D', 'C'): (5, 0),
    ('D', 'D'): (1, 1),
}
```

D. Implementasi Algoritma Keputusan

Keempat algoritma diimplementasikan dalam satu fungsi `get_move()` yang menerima parameter strategi, hasil interaksi node itu sendiri pada ronde sebelumnya, dan hasil interaksi lawan pada ronde sebelumnya:

```
def get_move(strategy, my_last,
opp_last):
    if strategy == 'AllD':
        return 'D'
    elif strategy == 'AllC':
        return 'C'
    elif strategy == 'TfT':
        if opp_last is None:
            return 'C'
        return opp_last[0]
    elif strategy == 'Pavlov':
        if my_last is None:
            return 'C'
        last_move, last_payoff = my_last
        if last_payoff >= 3:
            return last_move
    else:
        return 'D' if last_move ==
'C' else 'C'
```

Parameter `my_last` dan `opp_last` masing-masing berupa tuple (move, payoff) dari ronde sebelumnya, atau None apabila merupakan ronde pertama.

E. Loop Simulasi Utama

Simulasi dijalankan dengan menetapkan algoritma secara acak pada setiap node, kemudian melakukan iterasi sebanyak R ronde. Pada setiap ronde, seluruh sisi pada graf disimulasikan sebagai satu pertandingan Prisoner's Dilemma:

```
def run_simulation(G, num_rounds=100):
    strategies = ['AllD', 'AllC', 'TfT',
'Pavlov']
    node_strategy = {v:
random.choice(strategies) for v in
G.nodes()}
    node_score = {v: 0.0 for v in
G.nodes()}
    last_state = {v: None for v in
G.nodes()}

    for _ in range(num_rounds):
        new_state = {v: None for v in
G.nodes()}
        for u, v in G.edges():
            move_u =
get_move(node_strategy[u], last_state[u],
last_state[v])
            move_v =
get_move(node_strategy[v], last_state[v],
last_state[u])
            pay_u, pay_v =
PAYOFF[(move_u, move_v)]
            node_score[u] += pay_u
            node_score[v] += pay_v
```

```

        new_state[u] = (move_u,
pay_u)
        new_state[v] = (move_v,
pay_v)
        last_state = new_state

        return node_strategy, node_score

```

Implementasi ini hanya menyimpan hasil interaksi pada ronde terakhir (*last_state*) untuk setiap node, bukan seluruh riwayat interaksi. Hal ini cukup karena baik Tft maupun Pavlov hanya membutuhkan informasi satu ronde ke belakang dalam pengambilan keputusan, sehingga pendekatan ini secara signifikan mengurangi kompleksitas memori dan waktu komputasi dibandingkan menyimpan seluruh riwayat interaksi.

F. Pengukuran Properti Struktural Graf

Properti struktural setiap topologi diukur menggunakan fungsi bawaan NetworkX sebelum simulasi dijalankan:

```

def graph_properties(G):
    avg_degree =
sum(dict(G.degree()).values()) /
G.number_of_nodes()
    avg_clustering =
nx.average_clustering(G)
    return avg_degree, avg_clustering

```

G. Parameter Eksekusi

Untuk memperoleh hasil yang andal, setiap topologi disimulasikan sebanyak 5 kali pengulangan (*run*), dengan hasil akhir berupa rata-rata payoff per algoritma dari seluruh pengulangan. Graf sintetis dan Karate Club dijalankan sebanyak 10 run dengan 100 ronde per run, sementara ego-Facebook dijalankan sebanyak 3 run dengan 20 ronde per run mengingat ukuran graf yang jauh lebih besar (88.234 sisi).

V. HASIL DAN ANALISIS

A. Hasil

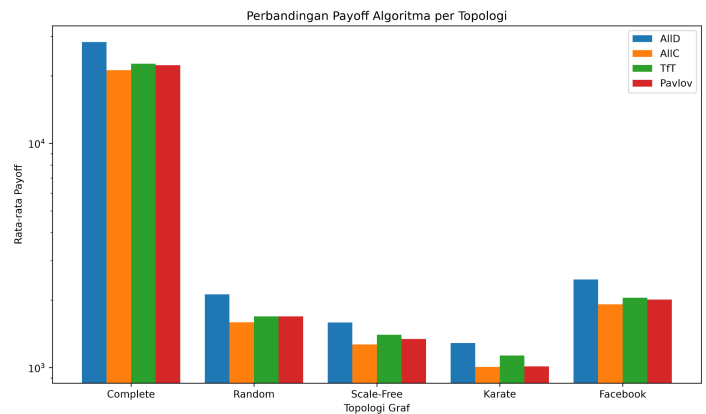
Eksekusi simulasi dilakukan menggunakan python dan dilakukan total lima kali eksekusi.

Keseluruhan proses ini diulang sebanyak 5 kali eksekusi independen untuk mengonfirmasi konsistensi temuan, dan Tabel 3 menyajikan rata-rata dari kelima eksekusi tersebut. Berikut adalah hasil rerata dari kelima simulasi tersebut:

Tabel 3. Hasil Rata-Rata Payoff dari Lima Kali Eksekusi

Topologi	AlID	AlIC	Tft	Pavlov	Pemenang
Complete	28.29 9	21.180	22.65 7	22.299	AlID
Random	2.120	1.592	1.692	1.694	AlID
Scale-Free	1.589	1.269	1.400	1.342	AlID
Karate	1.287	1.007	1.133	1.013	AlID
Facebook	2.473	1.916	2.050	2.012	AlID

Gambar 1. Grafik Payoff



Topologi yang digunakan selain *complete graph*, *random graph*, dan *scale-free graph* adalah karate dan facebook. Topologi Karate Club merupakan dataset bawaan library NetworkX berdasarkan penelitian Zachary (1977) dan topologi facebook merupakan database pertemanan facebook yang bisa diunduh di internet. Hasil akhir yang didapat adalah algoritma AlID (Always Defect) mendominasi di kelima run di semua topologi.

Hasil akhir yang didapat adalah algoritma AlID mendominasi di kelima eksekusi pada semua topologi, dengan gap keunggulan yang bervariasi antar topologi. 24,9% (complete), 25,2% (random), 13,5%(scale-free), 13,6%(karate), dan 20,6%(facebook).

Sebagai catatan tambahan, algoritma Tft (Tit for Tat) sempat mengalahkan AlID di eksekusi ke-3 pada topologi karate dengan selisih 65 poin dan di eksekusi ke-4 pada topologi scale-free dengan selisih 73 poin.

B. Analisis

Dari hasil simulasi, dapat ditemukan beberapa hal:

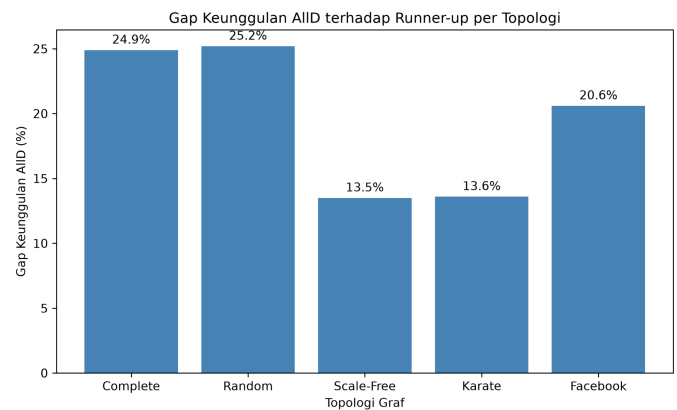
1. Analisis kemenangan tipis Tft

Dari lima eksekusi simulasi, ditemukan dua kasus di mana algoritma Tft berhasil mengalahkan AllD pada eksekusi ke-3 topologi Karate (Tft=1.217, AllD=1.152, selisih 65 poin) dan eksekusi ke-4 topologi Scale-Free (Tft=1.568, AllD=1.494, selisih 74 poin). Menariknya, kedua kemenangan tersebut hanya terjadi pada Tft, bukan AllC maupun Pavlov. Hal ini dapat dijelaskan melalui karakteristik masing-masing topologi. Pada Scale-Free, struktur hub memungkinkan node Tft yang menempati posisi hub untuk berinteraksi dengan sangat banyak tetangga sekaligus. Ketika hub tersebut berhadapan dengan mayoritas kooperator di sekitarnya, Tft mengakumulasi poin besar dari interaksi (C,C) yang berulang. Pada Karate Club, clustering coefficient yang tinggi (0,571) membuat node berinteraksi berulang kali dengan tetangga yang sama sehingga kondisi ini sangat menguntungkan Tft karena memori satu ronde ke belakang menjadi jauh lebih bermakna dibanding di topologi dengan koneksi yang lebih acak. Kedua kemenangan ini bersifat probabilistik yang sebenarnya bergantung pada distribusi acak strategi di awal simulasi. Namun kehadirannya di dua topologi yang sama (Scale-Free dan Karate) bukan kebetulan, melainkan mencerminkan bahwa karakteristik struktural kedua topologi tersebut memang lebih kondusif bagi algoritma berbasis memori. Selain itu, jumlah node yang relatif sedikit pada topologi scale-free serta karate juga berpengaruh meningkatkan std kedua topologi tersebut.

2. Dominasi AllD

Algoritma AllD terlihat jelas mendominasi di setiap run, tetapi keunggulannya tidak sama di setiap topologi. Gap keunggulan AllD terhadap runner-up-nya adalah 24,9% (Complete), 25,2% (Random), 13,5% (Scale-Free), 13,6% (Karate), dan 20,6% (Facebook). Hal ini mengindikasikan bahwa topologi mempengaruhi derajat keunggulan AllD meskipun tidak mengubah pemenangnya. Dominasi AllD juga diperkuat oleh tidak adanya mekanisme evolusi strategi dalam simulasi ini — node tidak dapat mengganti strategi meskipun terus kalah, sehingga AllD bebas mengeksploitasi tanpa konsekuensi jangka panjang.

Gambar 2. Grafik Gap Keunggulan AllD



3. Hubungan avg_degree dengan skor absolut

Ini merupakan hubungan matematis. Skor sebuah node merupakan jumlah payoff dari semua interaksinya. Interaksi per ronde sama dengan jumlah tetangga yang juga sama dengan degree node tersebut. Alhasil ditemukan bahwa skor berbanding lurus dengan hasil kali degree, payoff per interaksi, dan jumlah ronde.

4. Topologi paling stabil

Topologi Facebook mencatat standar deviasi paling rendah di antara semua topologi (AllD std=24,4 pada eksekusi pertama). Stabilitas ini disebabkan oleh jumlah node yang besar (4.039 node) sehingga efek acak dari penempatan strategi mampu ter-average dengan baik, berbeda dengan Karate Club yang hanya memiliki 34 node sehingga satu node AllD yang kebetulan terhubung ke banyak AllC dapat mengubah hasil secara signifikan.

Berdasarkan seluruh hasil di atas, dapat disimpulkan bahwa topologi graf mempengaruhi performa algoritma keputusan, meskipun tidak mengubah algoritma dominan. Pengaruh topologi terlihat pada tiga aspek: (1) skor absolut yang berbanding lurus dengan avg_degree, (2) derajat keunggulan AllD yang bervariasi antar topologi, dan (3) kemunculan kemenangan Tft pada topologi dengan karakteristik struktural tertentu.

IV. KESIMPULAN

Berdasarkan hasil simulasi, dapat disimpulkan bahwa topologi graf mempengaruhi performa algoritma keputusan berbasis memori dalam simulasi Iterated Prisoner's Dilemma, meskipun tidak mengubah algoritma yang mendominasi, dalam kasus ini adalah AllD. Pengaruh topologi terlihat pada tiga aspek utama yakni skor absolut yang berbanding lurus dengan rata-rata degree, derajat keunggulan AllD yang bervariasi antar topologi, serta kemunculan kemenangan Tft pada topologi Scale-Free dan Karate Club yang memiliki karakteristik struktural kondusif bagi algoritma berbasis memori. Stabilitas hasil simulasi juga dipengaruhi oleh topologi. Topologi dengan jumlah interaksi total yang besar

seperti topologi ego-Facebook menghasilkan standar deviasi rendah karena efek acak ter-average dengan baik.

Dalam simulasi ini, AllD mendominasi seluruh topologi. Hasil ini berbeda dari turnamen Axelrod (1980) yang menggunakan format round-robin tanpa struktur jaringan atau topologi, di mana setiap strategi bertanding melawan semua strategi lain dan Tft unggul akibat akumulasi skor tertinggi secara keseluruhan. Perbedaan mekanisme inilah yang menjelaskan perbedaan hasil pada simulasi berbasis jaringan dengan strategi tetap sehingga AllD bebas mengeksploitasi tanpa konsekuensi karena tidak terdapat mekanisme evolusi strategi. Tanpa adanya evolusi strategi, kooperator tidak dapat membentuk cluster pertahanan yang dalam literatur Nowak & May (1992), terbukti mampu membatasi dominasi AllD pada topologi dengan clustering coefficient tinggi.

Penelitian lanjutan dapat mengimplementasikan mekanisme evolusi strategi, di mana node yang kalah mengadopsi strategi tetangga yang lebih sukses, untuk menguji apakah dominasi AllD berkurang secara signifikan, khususnya pada topologi ego-Facebook (clustering=0,606) dan Karate Club (clustering = 0,571) yang secara teoritis paling kondusif untuk kemunculan kooperasi spontan.

REFERENCES

- [1] R. Axelrod, *The Evolution of Cooperation*. New York: Basic Books, 1984.
- [2] M. A. Nowak and R. M. May, "Evolutionary games and spatial chaos," *Nature*, vol. 359, pp. 826–829, 1992.
- [3] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.
- [4] P. Erdős and A. Rényi, "On random graphs I," *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.
- [5] J. Leskovec and R. Krevl, "SNAP Datasets: Stanford Large Network Dataset Collection," <http://snap.stanford.edu/data>, Jun. 2014.
- [6] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.

[7] Rinaldi Munir, "Graf," *Bahan Kuliah IF1220 Matematika Diskrit*, STEI-ITB, 2026.

[8] A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th Python in Science Conf. (SciPy 2008)*, 2008, pp. 11–15.

Program yang dipakai untuk penelitian paper ini bisa dilihat melalui tautan berikut:

<https://github.com/codenameskyfall/matdis-IPD-simulation>

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, 18 Juni 2026



Moch Naufal Zaki Basara
13525095