Desain Algoritma Rekomendasi Harga Dinamis pada Jaringan Pedagang Pasar Tradisional Berbasis Teori Graf

Niko Samuel Simanjuntak - 13524029
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: nikosimanjuntak71@gmail.com, 13524029@std.stei.itb.ac.id

***Abstrak*—Pasar tradisional merupakan salah satu bentuk perdagangan yang memiliki karakteristik unik dalam pembentukan harga barang, dengan interaksi antarpedagang memengaruhi dinamika harga secara lokal. Dalam konteks ini, penting untuk merancang sistem rekomendasi harga yang adaptif dan relevan dengan kondisi lingkungan sosial pasar. Makalah ini mengusulkan sebuah algoritma rekomendasi harga dinamis berbasis teori graf, dengan memodelkan pedagang sebagai simpul dan hubungan kompetitif atau kolaboratif antar pedagang sebagai sisi pada graf berbobot. Algoritma ini menggunakan pendekatan rerata harga tetangga untuk menyarankan harga ideal secara lokal dan terdistribusi. Melalui simulasi sederhana pada jaringan graf kecil, algoritma ini menunjukkan potensi untuk menyebarkan harga secara konsisten dan efisien di antara simpul yang saling terhubung. Pendekatan ini diharapkan dapat menjadi dasar bagi pengembangan sistem rekomendasi harga yang adil, transparan, dan berkelanjutan dalam ekosistem pasar tradisional.**

Kata Kunci—graf; pasar tradisional; rekomendasi harga; harga dinamis; jaringan pedagang

I. PENDAHULUAN

Pasar tradisional memiliki peran penting dalam mendistribusikan kebutuhan pokok masyarakat. Tidak seperti sistem harga tetap di ritel modern, harga di pasar tradisional bersifat fleksibel dan sangat dipengaruhi oleh interaksi antar pedagang, lokasi, dan ketersediaan barang. Dalam praktiknya, pedagang seringkali menyesuaikan harga dengan melihat harga dari pedagang di sekitarnya atau berdasarkan informasi informal yang menyebar di lingkungan pasar.

Namun, belum banyak pendekatan sistematis atau algoritmik yang dirancang untuk membantu pedagang dalam menentukan harga jual yang kompetitif dan adil secara adaptif. Salah satu pendekatan yang potensial adalah dengan menggunakan teori graf untuk merepresentasikan struktur hubungan antarpedagang dalam sebuah jaringan. Dalam model ini, setiap pedagang direpresentasikan sebagai simpul, dan hubungan—baik berupa kedekatan lokasi dan jenis barang dagangan—direpresentasikan sebagai sisi berbobot pada graf.

Makalah ini bertujuan untuk merancang algoritma sederhana berbasis teori graf yang dapat merekomendasikan harga dinamis bagi pedagang dalam jaringan tersebut. Dengan memanfaatkan prinsip penyebaran informasi lokal, seperti rata-rata harga tetangga yang dimodifikasi, algoritma ini diharapkan mampu memberikan rekomendasi harga yang rasional dan konvergen di seluruh jaringan pedagang.

Kontribusi utama dari makalah ini adalah perancangan model graf pasar tradisional dan formulasi algoritma rekomendasi harga dinamis yang berbasis propagasi lokal, yang sederhana namun efektif untuk diterapkan. Pendekatan ini juga membuka peluang untuk pengembangan sistem pendukung keputusan harga yang terdesentralisasi di masa depan.

II. DASAR TEORI

A. Teori Graf

Graf adalah struktur data yang terdiri dari sekumpulan titik (disebut simpul atau *node*) dan garis yang menghubungkan pasangan titik tersebut (disebut sisi atau *edge*). Graf digunakan untuk merepresentasikan objek-objek dan hubungan antarobjek tersebut, terutama ketika hubungan itu bersifat diskrit, misalnya hubungan pertemanan, jaringan jalan, atau koneksi antar komputer.

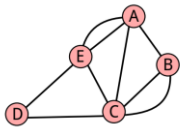
1) Definisi Graf

Sebuah graf G didefinisikan sebagai pasangan terurut

$$G = (V, E)$$

dengan

- V adalah himpunan tidak-kosong simpul (*vertices*), $V = \{v_1, v_2, \ldots, v_n\}$, dan
- E adalah himpunan sisi (*edges*) yang menghubungkan pasangan simpul, $E = \{e_1, e_2, \ldots, e_n\}$.



Gambar 1. Simpul direpresentasikan dengan bentuk lingkaran dan sisi direpresentasikan dengan bentuk garis [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

2) Jenis Graf

Hubungan antarsimpul dalam sebuah graf dapat diklasifikasikan berdasarkan dua aspek, yaitu arah sisi dan keberadaan bobot pada sisi tersebut.

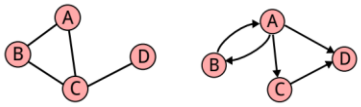
a) Berdasarkan Arah Sisi

- Graf Tak Berarah

  Pada graf tak berarah, setiap sisi antarsimpul bersifat dua arah. Artinya, jika terdapat sebuah sisi yang menghubungkan simpul A dan B, maka lintasan dapat dilakukan baik dari A ke B maupun dari B ke A tanpa perbedaan arah.

- Graf Berarah

  Pada graf berarah, setiap sisi memiliki arah tertentu. Jika terdapat sisi dari A ke B, maka lintasan hanya bisa dilakukan dari A menuju B, dan bukan sebaliknya, kecuali terdapat sisi terpisah dari B ke A.



Gambar 2. Graf tak berarah (kiri) dan graf berarah (kanan) [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

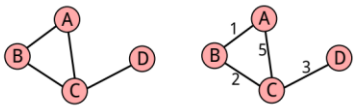
b) Berdasarkan Keberadaan Bobot

- Graf Tak Berbobot

  Graf tak berbobot memiliki sisi yang tidak memiliki nilai bobot tertentu. Setiap koneksi hanya menunjukkan keberadaan hubungan antarsimpul, tanpa mempertimbangkan metrik apapun.

- Graf Berbobot

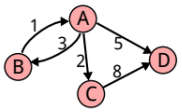
  Pada graf berbobot, setiap sisi memiliki nilai tertentu yang merepresentasikan ukuran atau biaya dari hubungan tersebut yang relevan tergantung konteks penggunaan graf.



Gambar 3. Graf tak berbobot (kiri) dan graf berbobot (kanan) [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

c) Kombinasi Keberadaan Arah dan Bobot

Sebuah graf dapat memiliki arah pada setiap sisi sekaligus memberikan bobot pada masing-masing sisi tersebut.



Gambar 4. Graf berbobot dan berarah [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

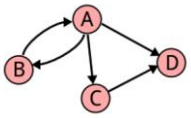
3) Representasi Graf

Graf dapat direpresentasikan dalam berbagai cara tergantung pada kebutuhan analisis dan efisiensi ruang atau waktu.

a) Matriks Ketetanggaan (Adjacency Matrix)

Pada representasi ini, digunakan matriks M dengan ukuran $V \times V$, dengan V merupakan banyaknya simpul.

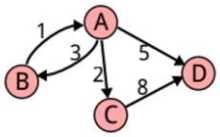
- Pada graf tidak berbobot, $M_{ij} = 1$ jika terdapat sisi dari i ke j dan $M_{ij} = 0$ jika tidak ada.



	A	B	C	D
A	0	1	1	1
B	1	0	0	0
C	0	0	0	1
D	0	0	0	0

Gambar 5. Adjacency matrix graf tidak berbobot [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

- Pada graf berbobot, $M_{ij} = w$ jika terdapat sisi dari i ke j dengan bobot w dan $M_{ij} = \infty$ jika tidak ada.



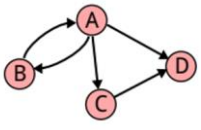
	A	B	C	D
A	∞	3	2	5
B	1	∞	∞	∞
C	∞	∞	∞	8
D	∞	∞	∞	∞

Gambar 6. Adjacency matrix graf berbobot [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

b) Daftar Ketetanggaan (Adjacency List)

Pada representasi ini, untuk setiap simpul, disiapkan sebuah struktur penyimpanan yang mencatat informasi mengenai simpul-simpul yang terhubung langsung dengannya.

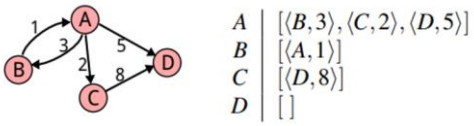
- Pada graf tidak berbobot, cukup disimpan daftar simpul tetangga untuk setiap simpul.



A	[B,C,D]
B	[A]
C	[D]
D	[ ]

Gambar 7. Adjacency list graf tidak berbobot [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

- Sedangkan pada graf berbobot, selain simpul tetangga, dicatat juga bobot dari setiap sisi yang menghubungkannya.

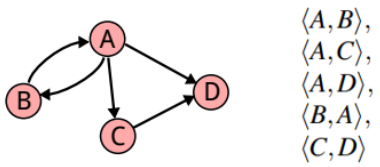


Gambar 8. Adjacency list graf berbobot [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

c) Daftar Sisi (Edge List)

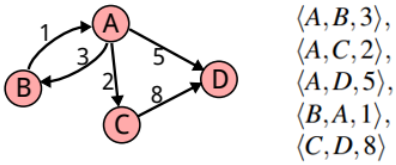
Pada representasi ini, semua informasi mengenai sisi disimpan dalam sebuah penampung yang berisi daftar pasangan simpul. Setiap elemen dalam daftar tersebut mewakili satu sisi dalam graf.

- Pada graf tidak berbobot, setiap sisi direpresentasikan sebagai pasangan simpul yang terhubung.



Gambar 9. Edge list graf tidak berbobot [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

- Pada graf berbobot, setiap sisi direpresentasikan sebagai tripel yang mencakup dua simpul dan bobot dari sisi tersebut.



Gambar 10. Edge list graf berbobot [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

4) Tetangga dan Lingkungan Simpul

Neighborhood dari simpul *i* dinotasikan *N*(*i*), yaitu himpunan simpul yang terhubung langsung ke *i*:

$$N(i) = \{j \in V \mid (i,j) \in E\}$$

B. Algoritma Rata-Rata Tetangga (Weighted Neighbor Averaging)

1) Prinsip Konsensus Linier

Algoritma rekomendasi harga yang digunakan mengikuti prinsip *distributed linear consensus*. Dalam konteks ini, setiap pedagang (simpul) menyesuaikan harga jualnya secara iteratif dengan memperhatikan harga tetangga-tetangganya, berdasarkan bobot kedekatan. Pembaruan harga pada iterasi ke-*t* + 1 dirumuskan sebagai:

$$P_i^{(t+1)} = P_i^{(t)} + \epsilon \sum_{j \in N(i)} w_{ij} \, (P_j^{(t)} - P_i^{(t)}),$$

dengan:

- $P_i^{(t)}$: harga pada simpul *i* di iterasi ke-*t*,
- w_{ij} : bobot sisi antara simpul *i* dan *j*,
- ϵ : faktor peredam atau *step size* ($0 < \epsilon \leq 1$),
- $N(i)$: himpunan tetangga dari simpul *i*.

Skema ini merepresentasikan bahwa setiap pedagang menggeser harga jualnya mendekati rata-rata tertimbang dari harga para tetangganya.

2) Parameter Damping (ϵ)

Faktor ϵ berperan sebagai pengatur laju penyesuaian harga. Nilai ϵ yang kecil (< 0.3) membuat konvergensi lambat tetapi stabil, sedangkan nilai besar (mendekati 1) dapat mempercepat proses, namun berisiko menyebabkan fluktuasi atau osilasi jika topologi graf tidak stabil.

3) Kriteria Konvergensi

Algoritma iteratif ini berjalan hingga perubahan harga pada seluruh simpul berada di bawah ambang batas tertentu (δ). Kriteria berhenti ditentukan dengan menghitung selisih maksimum antar iterasi:

$$max_{i \in V} \left| P_i^{(t+1)} - P_i^{(t)} \right| < \delta$$

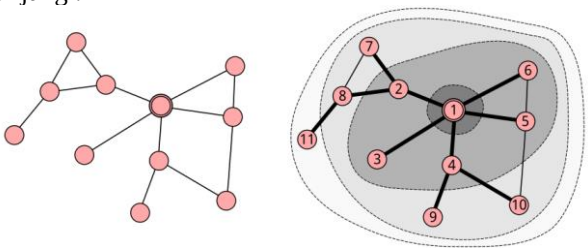
Nilai δ yang digunakan kecil, seperti 10^{-3}, untuk menjamin bahwa harga antarpedagang telah cukup stabil.

4) Properti Konvergensi

Berdasarkan teori konsensus linier, jika graf *G* terhubung dan bobot w_{ij} > untuk semua $(i, j) \in E$, maka algoritma ini akan konvergen. Nilai akhir P_i untuk semua simpul *i* dalam satu komponen terhubung akan mendekati rata-rata tertimbang dari harga awal simpul-simpul dalam komponen tersebut. Dengan demikian, hasil akhirnya bergantung pada struktur graf dan keterhubungan antarsimpul.

C. Algoritma Breadth-First Search (BFS)

Breadth-First Search (BFS) merupakan salah satu algoritma untuk menjelajahi graf dengan menelusuri simpul-simpul berdasarkan aturan tertentu. Dalam implementasinya, BFS menggunakan struktur data antrian (*queue*) untuk menyimpan daftar simpul yang akan dieksplorasi. Pada setiap langkah, simpul yang berada di depan antrian akan diambil untuk dikunjungi, kemudian seluruh simpul tetangganya yang belum dikunjungi dimasukkan ke bagian belakang antrian. Proses ini berlanjut hingga seluruh simpul pada graf telah dikunjungi.



Gambar 11. Graf dan urutan pengunjungan node -nya secara BFS [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

Algoritma 45 Penelusuran graf dengan BFS.

```
1:  procedure BFS(initialNode)
2:      INITIALIZEQUEUE()                          ▷ Inisialisasi sebuah queue kosong.
3:      PUSH(initialNode)
4:      visited[initialNode] ← true
5:      while not ISEMPTYQUEUE() do ▷ ISEMPTYQUEUE: bernilai true jika queue kosong.
6:          curNode ← FRONT()
7:          POP()
8:          print "mengunjungi curNode"
9:          for adjNode ∈ adj(curNode) do
10:             if not visited[adjNode] then
11:                 PUSH(adjNode)
12:                 visited[adjNode] ← true
13:             end if
14:         end for
15:     end while
16: end procedure
```

Gambar 12. Pseudocode algoritma BFS [Sumber: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf]

D. Simulasi Spasial dan Graf Unit-Disk

1) Graf Geometrik (Unit-Disk Graph)

Dalam simulasi, keterhubungan antarpedagang didasarkan pada jarak spasial antarlokasi mereka di pasar. Dua simpul *i* dan *j* akan terhubung dalam graf jika jarak *Euclidean* mereka lebih kecil atau sama dengan radius ambang *r*:

$$dist(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r.$$

Graf dengan aturan ini disebut sebagai *unit-disk graph*, lazim digunakan untuk memodelkan jaringan spasial

2) Aplikasi dalam Pasar Tradisional

Model graf geometrik ini sangat sesuai untuk pasar tradisional, karena interaksi harga di antara pedagang cenderung terjadi secara lokal. Pedagang yang berlokasi berdekatan cenderung saling mengamati harga satu sama lain dan menyesuaikan harga jual secara informal. Oleh karena itu, pembentukan graf berbasis jarak fisik mencerminkan struktur alami jaringan informasi harga di lingkungan pasar tersebut.

III. IMPLEMENTASI ALGORITMA

A. Pemodelan Jaringan Pedagang

Dalam konteks pasar tradisional, jaringan pedagang dimodelkan sebagai jaringan spasial dengan setiap pedagang direpresentasikan sebagai simpul dalam graf dan interaksi antarpedagang diwakili oleh sisi antarsimpul. Posisi fisik pedagang di pasar (misalnya koordinat dua dimensi (x, y)) sangat memengaruhi kemungkinan terjadinya interaksi langsung, seperti pertukaran informasi harga atau kompetisi produk. Dengan pendekatan graf tak berarah berbobot, struktur pasar diringkas dalam bentuk jaringan berupa pedagang berdekatan cenderung saling terhubung, sedangkan yang berjauhan memiliki pengaruh yang lebih kecil. Pemodelan graf ini memungkinkan penggunaan algoritma graf untuk merekomendasikan harga dinamis berdasarkan struktur hubungan lokal di pasar.

Graf pedagang yang dihasilkan direpresentasikan menggunakan *adjacency list* (daftar ketetanggaan). Struktur ini dipilih karena efisien untuk graf jarang. *Adjacency list* hanya menyimpan simpul dan sisi yang benar-benar ada, sehingga hemat ruang memori dibandingkan *matrix adjacency* jika graf memiliki lebih banyak simpul daripada sisi.

Proses pembentukan sisi dalam graf pedagang didasarkan pada jarak *Euclidean* antarpedagang. Sebagai penyederhanaan, digunakan *threshold* jarak *r*, yaitu hanya pasangan pedagang dengan jarak $\leq r$ yang dianggap berinteraksi langsung (terhubung). Pendekatan ini mirip dengan konsep graf geometrik acak (*random geometric graph*), dengan dua simpul dihubungkan jika jarak di antara mereka kurang dari suatu parameter tertentu. Dengan demikian, jaringan pedagang bersifat lokal: pedagang yang secara geografis dekat berpeluang besar saling terhubung, sementara pedagang yang jauh secara otomatis tidak memiliki sisi langsung. Penggunaan *threshold* menjaga kepadatan graf tetap wajar dan menggambarkan keterbatasan interaksi jarak jauh. Untuk setiap pasangan pedagang (i, j) yang terhubung, bobot sisi $w(i, j)$ dihitung sebagai fungsi invers jarak, yaitu

$$w(i,j) = \frac{1}{1 + dist(i,j)}$$

dengan $dist(i, j)$ adalah jarak *Euclidean* antarpedagang. Fungsi ini menghasilkan bobot yang berbanding terbalik dengan jarak sehingga pedagang yang lebih dekat memiliki bobot koneksi lebih besar. Pendekatan pembobotan ini memodelkan bahwa pengaruh atau interaksi antarpedagang melemah seiring bertambahnya jarak fisik. Dengan kata lain, pedagang tetangga memiliki pengaruh harga yang lebih signifikan dibandingkan pedagang jauh, sesuai praktik bobot berbasis kedekatan dalam analisis jaringan spasial

Data input untuk simulasi disusun dalam format baris untuk setiap pedagang, yang mencakup informasi berikut:

- ID: Identitas unik pedagang (misalnya integer).
- Koordinat (x, y): Posisi pedagang dalam bidang dua dimensi (contoh: koordinat dalam satuan meter).
- Harga awal: Harga awal produk pedagang (nilai numerik, misalnya dalam rupiah).

Contoh satu baris data input: `[ID, x, y, harga_awal]`, misalnya `(1, 23.5, 47.2, 10000)` untuk pedagang dengan ID=1, koordinat (23.5, 47.2), dan harga awal 10000.

B. Perhitungan Rata-Rata Harga Tetangga Berbobot

Sebagai dasar algoritma rekomendasi harga dinamis, setiap simpul pedagang *i* diperbarui harga rekomendasinya berdasarkan rata-rata berbobot dari harga-harga tetangganya. Misalkan graf pasar tradisional berbobot adalah $G = (V, E)$ dengan simpul V (pedagang) dan sisi berbobot w_{ij} antara simpul *i* dan *j*. Definisikan $P_i^{(t)}$ sebagai harga rekomendasi pada simpul *i* pada iterasi ke-*t*, dan $N(i) = \{j \mid (i, j) \in E\}$ sebagai himpunan tetangga *i*. Secara formal, pada setiap iterasi ke-*t*, simpul *i* mengumpulkan informasi harga $P_j^{(t)}$ dari semua tetangganya $j \in N(i)$. Nilai baru $P_i^{(t+1)}$ dihitung dengan rumus rata-rata berbobot:

$$P_i^{(t+1)} = \frac{\sum_{j \in N(i)} w_{ij} P_j^{(t)}}{\sum_{j \in N(i)} w_{ij}} \; dengan \; w_{ij} \geq 0,$$

w_{ij} adalah bobot pengaruh tetangga *j* terhadap *i*. Dengan definisi ini, setiap simpul menggeser harga rekomendasinya menuju rata-rata harga lingkungan sekitar. Proses ini bersifat iteratif, dimulai dari kondisi awal $P_i^{(0)}$ (harga dasar), kemudian diperbarui berulang kali hingga mencapai kestabilan. Iterasi ini dapat dimodelkan sebagai $P^{(t+1)} = A\, P^{(t)}$ dengan matriks bobot *A*

berdimensi $|V| \times |V|$ dengan $A_{ij} = w_{ij} / \sum_{k \in N(i)} w_{ik}$. Bila graf terhubung dan bobot disusun tepat, maka iterasi ini dapat mendekati kesepakatan seragam atas harga konsensus di seluruh simpul.

C. Propagasi Harga Berdasarkan Urutan BFS

Untuk efektivitas implementasi, pembaruan harga dapat dilakukan mengikuti urutan jelajah BFS (*Breadth-First Search*). Algoritma BFS memulai dari satu simpul akar dan mengunjungi simpul-simpul menurut kedalaman lapisan graf. Dalam konteks ini, pemilihan simpul akar (misalnya penjual yang memulai perubahan harga) dan pelacakan BFS memastikan simpul terdekat (lapisan-lapisan pertama) diperbarui lebih dahulu, lalu diteruskan ke simpul berikutnya. Dengan demikian, perubahan harga merambat secara bertingkat ke seluruh jaringan. Secara prosedural, BFS menggunakan antrian (*queue*) untuk melacak simpul yang akan diproses selanjutnya. Skema iteratif digabung dengan BFS, yaitu pada setiap iterasi *t*, dijalankan satu siklus BFS untuk memperoleh urutan kunjungan simpul. Selanjutnya, harga baru $P_i^{(t+1)}$ dihitung sesuai tetangga (yang sebagian besar sudah ter-*visit*) dalam urutan BFS tersebut. Pendekatan ini menjamin setiap simpul akhirnya diperbarui (graf terhubung menjamin semua simpul ter-*cover*) dan penyebaran informasi harga terstruktur.

D. Pseudocode Algoritma

```
procedure WEIGHTEDNEIGHBORBFS(G, w, P0, ε, T_max)
  ▷ G = (V, E) graf berbobot; w[i,j] bobot pada sisi (i,j)
  ▷ P0[i] harga awal untuk setiap simpul i; ε toleransi; T_max iterasi maksimum
  t ← 0
  ▷ Inisialisasi P_old dengan nilai awal P0
  for i ← 1, |V| do
    P_old[i] ← P0[i]
  end for
  while t < T_max do
    ▷ Tandai semua simpul sebagai belum dikunjungi
    for i ← 1, |V| do
      visited[i] ← false
    end for
    ▷ Pilih akar s (contoh: s = 1)
    s ← 1
    visited[s] ← true
    ▷ Inisialisasi antrian Q dan enqueue s
    Q ← new queue
    enqueue(Q, s)
    ▷ Proses BFS dengan perhitungan harga bertahap
    while not empty(Q) do
      u ← dequeue(Q)
      ▷ Enqueue tetangga yang belum dikunjungi
      for each j ∈ N(u) do
        if visited[j] = false then
          visited[j] ← true
          enqueue(Q, j)
        end if
      end for
      ▷ Hitung harga P_new[u] berdasarkan tetangga pada iterasi t
      num ← 0
      den ← 0
      for each j ∈ N(u) do
        num ← num + w[u, j] * P_old[j]
        den ← den + w[u, j]
      end for
      if den ≠ 0 then
        P_new[u] ← num / den
      else
        P_new[u] ← P_old[u] ▷ jika tidak ada tetangga, pertahankan nilai lama
      end if
    end while
    ▷ Cek kriteria konvergensi: hitung perbedaan maksimum
    delta_max ← 0
    for i ← 1, |V| do
      delta ← abs(P_new[i] – P_old[i])
      if delta > delta_max then
        delta_max ← delta
      end if
    end for
    if delta_max < ε then
      break ▷ sudah konvergen
    end if
    ▷ Persiapan iterasi selanjutnya: salin P_new → P_old
    for i ← 1, |V| do
      P_old[i] ← P_new[i]
    end for
    t ← t + 1
  end while
  ▷ Kembalikan harga konvergen; P_final[i] di P_new[i]
  return P_new
end procedure
```

Gambar 13. Pseudocode algoritma WEIGHTEDNEIGHBORBFS [Sumber: Dokumentasi pribadi]

Keterangan simbol: $P_i^{(t)}$ adalah harga simpul *i* pada iterasi *t*, $N(u)$ himpunan tetangga *u*, $\varepsilon > 0$ toleransi perbedaan harga untuk berhenti, dan T_{max} batas iterasi. Setiap iterasi melibatkan satu kali jelajah BFS—dengan enqueue simpul akar dan meluas hingga semua simpul terproses—kemudian perhitungan harga berbobot sesuai rumus. Proses diulang hingga pergeseran harga antar iterasi cukup kecil ($< \varepsilon$) atau mencapai batas T_{max}.

E. Kompleksitas Waktu dan Ruang

Analisis kompleksitas menggunakan notasi Big-O dengan $n = |V|$ jumlah simpul dan $m = |E|$ jumlah sisi. Setiap iterasi utama melakukan:

- Satu kali traversal BFS pada graf, yang memerlukan waktu $O(n + m)$ karena setiap simpul dan setiap sisi diproses sekali.
- Pembaruan harga pada setiap simpul, yang untuk simpul *u* mengeksekusi penjumlahan sepanjang tetangganya $N(u)$. Total biaya ini sepanjang satu iterasi juga $O(\sum_u |N(u)|) = O(m)$.

Secara keseluruhan, satu iterasi memakan waktu $O(n + m)$. Jika algoritma dijalankan hingga *T* iterasi (atau sampai konvergensi), total waktu kumulatif $O(T(n + m))$. Ruang tambahan yang diperlukan meliputi struktur penyimpanan graf sebesar $O(n + m)$, larik harga P_i sebesar $O(n)$, dan ruang antrian BFS $O(n)$ (sebagian besar karena visited array). Dengan demikian, kompleksitas ruang total juga $O(n + m)$.

Pada kasus terburuk, misalkan graf padat ($m = O(n^2)$), maka kompleksitas waktu per iterasi $O(n^2)$. Namun pada umumnya jaringan pedagang di pasar tradisional lebih bersifat jarang (*sparse*), sehingga $m = O(n)$ dan komputasi per iterasi $O(n)$. Dalam praktiknya, iterasi dihentikan ketika perubahan harga relatif sudah kecil (berdasarkan toleransi ε) sehingga tidak selalu perlu mencapai T_{max}.

IV. STUDI KASUS DAN EVALUASI

Seluruh proses studi kasus berikut menggunakan implementasi Python interaktif, yang disusun dan dilampirkan secara lengkap dalam bagian Lampiran. Hal ini bertujuan agar simulasi dapat direproduksi dan dikaji ulang oleh pembaca.

A. *Desain Simulasi Jaringan Pedagang*

Pada tahap awal simulasi, pengguna memasukkan data interaktif berupa jumlah pedagang, ukuran area pasar, serta detail setiap pedagang (ID, posisi koordinat (*x*, *y*), dan harga awal). Mekanisme input interaktif ini dirancang untuk memberikan fleksibilitas dalam mengeksplorasi berbagai skenario pasar tradisional dengan mudah, misalnya mahasiswa atau peneliti dapat menentukan jumlah pedagang dan parameter-parameter yang berbeda untuk mengamati perilaku algoritma dalam berbagai kondisi. Parameter-parameter utama dalam simulasi ini meliputi:

- Radius konektivitas (*r*): ambang jarak maksimum antara dua pedagang agar dianggap saling berinteraksi. Jika jarak antara dua pedagang kurang dari atau sama dengan *r*, maka terbentuk sisi (*edge*) di antara mereka. Dengan cara ini, dibangun sebuah graf tidak berarah yang merepresentasikan jaringan pedagang.
- Toleransi konvergensi (*tolerance*): ambang batas perubahan harga rekomendasi terkecil agar algoritma berhenti iterasi. Jika selisih harga antar iterasi berada di bawah nilai ini untuk semua pedagang, dianggap telah konvergen.
- Bobot pembaruan (ε): faktor pembobot dalam perhitungan rata-rata tertimbang harga tetangga. Faktor ε ($0 < \varepsilon \leq 1$) berperan untuk mengendalikan seberapa cepat pedagang menyesuaikan harganya berdasarkan harga tetangga; nilai ε yang terlalu besar dapat membuat fluktuasi cepat (bahkan osilasi jika $\varepsilon = 1$ dan graf hanya memiliki dua simpul yang saling berdekatan), sedangkan nilai ε kecil memperlambat konvergensi.
- Maksimum iterasi: jumlah iterasi maksimal untuk memastikan simulasi berakhir jika konvergensi lambat terjadi.

Graf yang terbentuk pada langkah awal ini disimpan sebagai *adjacency list*, dengan setiap simpul (pedagang) menyimpan daftar tetangganya. Misalnya, adjList[i] berisi semua indeks pedagang yang bertetangga dengan pedagang. Setelah graf terbentuk, proses simulasi melanjutkan ke perhitungan rekomendasi harga secara iteratif berdasarkan harga tetangga setiap pedagang.

B. *Hasil Simulasi Konvergensi dan Distribusi Harga*

Sebagai ilustrasi, pertimbangkan contoh simulasi sederhana berikut. Digunakan masukan berupa 4 pedagang di dalam area 10 × 10 meter dengan parameter $r = 4.0$, $\varepsilon = 0.5$, dan toleransi konvergensi = 0.01. Data masukan pedagang adalah:

- Pedagang 1: posisi (2, 3), harga awal 100.
- Pedagang 2: posisi (5, 5), harga awal 120.
- Pedagang 3: posisi (8, 2), harga awal 80.
- Pedagang 4: posisi (9, 9), harga awal 150.

```
Masukkan jumlah pedagang (n): 4
Lebar area pasar (misal 100): 10
Tinggi area pasar (misal 100): 10
Threshold jarak tetangga (r): 4
Toleransi update harga ε (misal 0.5): 0.5
Toleransi konvergensi tol (misal 1e-3): 0.01
Batas iterasi maksimum: 10

Masukkan data pedagang:
ID, x, y, harga_awal (pisah koma): 1, 2, 3, 100
ID, x, y, harga_awal (pisah koma): 2, 5, 5, 120
ID, x, y, harga_awal (pisah koma): 3, 8, 2, 80
ID, x, y, harga_awal (pisah koma): 4, 9, 9, 150
```

Gambar 14. Proses pemasukan data ke dalam program [Sumber: Dokumentasi pribadi]

Dengan $r = 4.0$, terbentuk sisi hanya antara pedagang 1 dan 2 (jarak ≈ 3.6 m). Berikut daftar tetangga yang dicetak:

- Pedagang 1: tetangga = [2]
- Pedagang 2: tetangga = [1]
- Pedagang 3: tetangga = [] (tidak ada tetangga dalam radius)
- Pedagang 4: tetangga = []

Selanjutnya algoritma rata-rata tertimbang dijalankan iteratif. Pada iterasi pertama, pedagang 1 dan 2 saling menyesuaikan harga: harga baru 1 menjadi *100 + 0.5*(120–100) = 110*, dan pedagang 2 menjadi *120 + 0.5*(100–120) = 110*. Pedagang 3 dan 4 tetap karena tidak memiliki tetangga. Setelah iterasi kedua selisih perubahan di bawah toleransi, simulasi konvergen. Harga akhir yang direkomendasikan adalah:

- Pedagang 1: 110
- Pedagang 2: 110
- Pedagang 3: 80
- Pedagang 4: 150

```
Membangun koneksi tetangga (jarak ≤ r)...

Daftar tetangga tiap pedagang:
Pedagang 1: Tetangga = ['2']
Pedagang 2: Tetangga = ['1']
Pedagang 3: Tetangga = []
Pedagang 4: Tetangga = []

Konvergen pada iterasi ke-2 (max_diff=0.00000)

Harga rekomendasi akhir:
  Pedagang 1: 110.00
  Pedagang 2: 110.00
  Pedagang 3: 80.00
  Pedagang 4: 150.00
```

Gambar 15. Keluaran yang dihasilkan program [Sumber: Dokumentasi pribadi]

Contoh simulasi ini menggambarkan bagaimana harga tetangga saling berpengaruh hingga mencapai kesepakatan lokal pada komponen terhubung.

C. *Analisis Sensitivitas terhadap Parameter*

Parameter *r* (jarak *threshold*) sangat mempengaruhi topologi jaringan dan hasil akhir harga. Bila *r* lebih kecil (misalnya < 4 pada contoh di atas), jaringan dapat terfragmentasi sehingga pedagang terpisah hanya dipengaruhi oleh tetangga lokalnya. Sebaliknya, jika *r* lebih besar, lebih banyak sisi terbentuk dan jaringan menjadi lebih terhubung.

Misalnya, menaikkan *r* menjadi 6.0 pada contoh di atas menghubungkan semua pedagang (graf menjadi satu komponen terhubung). Dalam kasus ini seluruh pedagang akhirnya saling mempengaruhi dan harga akhir akan mendekati rata-rata global (sekitar 113 pada contoh). Dengan *r* kecil, harga akhir berbeda-beda per klaster (misalnya pedagang 3 dan 4 tetap pada harga awal karena terisolasi).

Gambar 16. Proses pemasukan data dan hasil keluaran program untuk r = 6 [Sumber: Dokumentasi pribadi]

Parameter ε juga krusial, jika ε sangat kecil (misal 0.1), perubahan harga per iterasi kecil sehingga konvergensi memakan iterasi lebih banyak. Bila ε besar mendekati 1, perubahan cepat namun berisiko terjadi fluktuasi atau osilasi jika struktur jaringan tidak stabil (terutama pada pasangan pedagang saja). Sebagai contoh lain, jika ε diset menjadi 1 pada dua pedagang yang saling terhubung, harga mereka akan saling bertukar nilai setiap iterasi dan tidak pernah stabil. Oleh karena itu, ε biasanya dipilih moderat ($0 < \varepsilon < 1$) untuk menyeimbangkan kecepatan dan kestabilan.

Toleransi konvergensi menentukan kapan iterasi dihentikan. Toleransi besar (misalnya 0.1) dapat menghentikan algoritma lebih cepat, tetapi harga akhir mungkin belum benar-benar konsisten di antara tetangga. Toleransi sangat kecil memaksa konvergensi lebih ketat, tetapi membutuhkan iterasi lebih banyak.

D. Efektivitas dan Keterbatasan

Pendekatan interaktif dan algoritma sederhana ini memiliki beberapa kelebihan. Pertama, desain input interaktif memudahkan pengguna bereksperimen dengan skenario yang berbeda—baik jumlah pedagang, distribusi posisi, maupun parameter algoritma—tanpa perlu mengubah kode dasar. Kedua, algoritma rekomendasi berbasis rata-rata tetangga bersifat *decentralized* dan ringkas, yaitu setiap pedagang hanya membutuhkan informasi harga tetangga terdekatnya dan melakukan perhitungan lokal sederhana. Pola ini sesuai dengan konsep *distributed consensus* pada *multi-agent*, dengan "tim agen mencapai kesepakatan terhadap variabel tertentu melalui interaksi dengan tetangganya. Pendekatan ini efisien dan menjamin terbangunnya kesepakatan harga jika jaringan pedagang saling terhubung dengan benar.

Meskipun demikian, terdapat keterbatasan signifikan. Algoritma ini mengasumsikan hubungan linier antara harga pedagang dan tetangganya (rata-rata tertimbang). Padahal dalam kenyataan pasar, harga dapat dipengaruhi faktor kompleks seperti permintaan global, strategi kompetitif, dan dinamika nonlinier lainnya. Selain itu, model jaringan dibangun statis berdasarkan jarak geografis saja (graf *unit-disk*). Hal ini tidak selalu mencerminkan koneksi komersial nyata (misalnya pelanggan atau pasokan bahan baku). Jika asumsi linier tersebut tidak valid, rekomendasi harga mungkin tidak akurat secara ekonomi. Topologi jaringan yang terfragmentasi juga mengakibatkan setiap komponen mencapai kesepakatan lokalnya sendiri dan tidak ada jaminan keselarasan harga di seluruh pasar. Dengan demikian, meski algoritma sederhana ini efektif dalam eksperimen berbasis graf, pengguna harus menyadari keterbatasan modelnya terkait asumsi dasar dan konteks pasar nyata.

V. KESIMPULAN

Makalah ini telah memperkenalkan sebuah algoritma rekomendasi harga dinamis untuk jaringan pedagang pasar tradisional yang dimodelkan menggunakan teori graf berbobot. Berdasarkan metodologi rata-rata tetangga berbobot dan mekanisme iteratif sederhana, setiap pedagang (simpul) hanya memerlukan informasi harga dari tetangganya untuk menyesuaikan harga rekomendasi secara lokal. Studi kasus interaktif dengan 4 pedagang menunjukkan bahwa:

- Graf interaktif yang dibangun dari input pengguna (ID, koordinat, harga awal) efektif memodelkan struktur pasar berbasis jarak fisik.
- Algoritma konsensus rata-rata terbukti konvergen pada semua komponen terhubung—harga akhir pada tiap klaster graf mendekati rata-rata harga awal nyata.
- Parameter seperti radius konektivitas (*r*), laju penyesuaian (ε), dan toleransi konvergensi memengaruhi kecepatan dan sifat konvergensi, namun algoritma relatif *robust* terhadap variasi nilai asumsi.
- Keterbatasan utama terletak pada asumsi linieritas propagasi harga dan topologi graf statis; dinamika permintaan, penawaran, dan faktor non-geografis lain belum terakomodasi.

Dengan demikian, pendekatan ini layak dijadikan kerangka dasar sistem rekomendasi harga terdesentralisasi untuk pasar tradisional sederhana, terutama jika konektivitas pedagang dapat dipetakan.

Berdasarkan hasil dan keterbatasan di atas, beberapa rekomendasi untuk penelitian lanjutan adalah:

- Graf Dinamis: Memperhitungkan perubahan topologi (pedagang baru/keluar, fluktuasi konektivitas) agar model merefleksikan dinamika pasar nyata.
- Model Non-Linier: Mengintegrasikan fungsi respons harga nonlinier atau model permintaan-penawaran untuk menangkap strategi penetapan harga yang lebih kompleks.
- Pembobotan Multi-Faktor: Menambahkan bobot berdasarkan volume transaksi, reputasi pedagang, atau jenis produk, bukan hanya jarak fisik.
- *Hybrid* dengan *Machine Learning*: Menggabungkan algoritma konsensus graf dengan teknik pembelajaran (misal *reinforcement learning*) untuk menyesuaikan parameter ε atau merancang fungsi manfaat pedagang.

Kode Python interaktif lengkap disertakan dalam Lampiran daring untuk memudahkan reproduksi dan eksplorasi lebih lanjut oleh pembaca.

VI. Lampiran

Seluruh kode Python yang digunakan dalam perancangan dan simulasi algoritma ini disusun dalam satu notebook Google Colab berjudul *simulasi_rekomendasi_harga_pedagang_graf.ipynb* dan dapat diakses oleh pembaca melalui tautan berikut: https://colab.research.google.com/drive/1cQf66pYdSIpGMFwRrbewZzh41_Y6LGF_

Ucapan Terima Kasih

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan kekuatan yang telah diberikan, sehingga penulis dapat menyelesaikan makalah ini sebagai bagian dari proses pembelajaran di semester kedua. Penulis juga menyampaikan terima kasih yang sebesar-besarnya kepada keluarga tercinta yang senantiasa memberikan dukungan moril, semangat, dan doa di setiap langkah perjalanan akademik ini. Ucapan terima kasih yang tulus juga penulis sampaikan kepada dosen pengampu mata kuliah IF1220 Matematika Diskrit K01, Bapak Dr. Ir. Rinaldi, M.T., atas bimbingan, materi yang inspiratif, dan ruang eksplorasi yang beliau berikan sepanjang perkuliahan. Tak lupa, apresiasi setinggi-tingginya juga penulis berikan kepada teman-teman seperjuangan yang selalu saling membantu, berbagi semangat, dan menjadi tempat bertukar pikiran di tengah segala tantangan yang dihadapi. Semoga segala kebaikan dan dukungan yang telah diberikan mendapatkan balasan yang setimpal.

Referensi

[1] K. Dewi and A. Nurhayati, "Pengaruh Harga, Kualitas, Kondisi Pasar dan Lokasi Pasar terhadap Preferensi Konsumen dalam Berbelanja di Pasar Tradisional," *Eqien*, vol. 3, no. 2, 2016. [Online]. Available: https://www.neliti.com/id/publications/282000/pengaruh-harga-kualitas-kondisi-pasar-dan-lokasi-pasar-terhadap-preferensi-konsu. [Accessed: Jun. 16, 2025].

[2] TOKI, "Pemrograman Kompetitif Dasar," OSN TOKI. [Online]. Available: https://osn.toki.id/data/pemrograman-kompetitif-dasar.pdf. [Accessed: Jun. 17, 2025].

[3] R. Munir, "Graf (Bagian 1)," Informatika STEI ITB, Bandung, Indonesia. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf. [Accessed: Jun. 17, 2025].

[4] R. Munir, "Graf (Bagian 2)," Informatika STEI ITB, Bandung, Indonesia. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf. [Accessed: Jun. 17, 2025].

[5] J. Diaz, D. Mitsche, and X. Perez, "Dynamic Random Geometric Graphs," arXiv preprint cs/0702074, 2007. [Online]. Available: https://arxiv.org/abs/cs/0702074. [Accessed: Jun. 18, 2025].

[6] D. Hallac, J. Leskovec, and S. Boyd, "Network Lasso: Clustering and Optimization in Large Graphs," in *Proc. 21st ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Sydney, NSW, Australia, 2015, pp. 387-396. [Online]. Available: https://web.stanford.edu/~boyd/papers/pdf/network_lasso.pdf. [Accessed: Jun. 19, 2025].

[7] S. Sundaram and C. N. Hadjicostis, "Distributed Function Calculation and Consensus using Linear Iterative Strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 748-764, Aug. 2011. [Online]. Available: https://engineering.purdue.edu/~sundara2/papers/journals/obs_func_calc_SunHad.pdf. [Accessed: Jun. 19, 2025].

[8] F. Pedroche, M. Rebollo, C. Carrascosa, and A. Palomares, "On the convergence of weighted-average consensus," *arXiv preprint arXiv:1307.7562*, 2013. [Online]. Available: https://arxiv.org/pdf/1307.7562. [Accessed: Jun. 19, 2025].

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, 20 Juni 2025

Niko Samuel Simanjuntak
13524029