

Perancangan Rute Shuttle Mahasiswa ITB Jatinangor Menggunakan Sirkuit Hamilton pada Graf Berbobot Berarah

Avicenna Ananda Musthafa - 13525018

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: avismusthafa1601@gmail.com, 13525018@std.stei.itb.ac.id

Abstract—Mobilitas mahasiswa ITB Jatinangor yang tinggal di kos sekitar kampus belum terlayani oleh sistem angkutan khusus, sementara lokasi hunian mahasiswa tersebar pada beberapa kawasan dengan arah jalan dan jarak tempuh yang tidak selalu simetris. Makalah ini merancang model konseptual rute shuttle mahasiswa dengan merepresentasikan ITB Jatinangor dan delapan kawasan kos sebagai graf berbobot berarah. Persoalan pencarian rute dimodelkan sebagai Asymmetric Travelling Salesman Problem (ATSP), yaitu pencarian sirkuit Hamilton berbobot minimum yang berangkat dari kampus, mengunjungi setiap kawasan tepat satu kali, lalu kembali ke kampus. Dua pendekatan algoritmik digunakan, yaitu Nearest Neighbor untuk memperoleh rute secara cepat berdasarkan pilihan titik terdekat, dan Brute Force untuk memperoleh rute optimal dengan mengevaluasi seluruh kemungkinan sirkuit. Eksperimen dilakukan pada dua skenario layanan, yaitu rute tunggal untuk seluruh kawasan dan dua rute terbagi berdasarkan kedekatan geografis. Hasil menunjukkan bahwa Skenario A menghasilkan total jarak minimum 13,25 km, sedangkan Skenario B menghasilkan total gabungan 15,40 km. Namun, Skenario B menurunkan panjang rute terpanjang per armada dari 13,25 km menjadi 8,50 km, atau turun sekitar 35,85%. Selain itu, 32 dari 36 pasangan lokasi memiliki jarak dua arah yang berbeda, sehingga pendekatan graf berarah terbukti penting. Dengan demikian, rute tunggal unggul untuk minimisasi jarak operasional, sedangkan rute terbagi lebih sesuai bila tujuan layanan adalah mengurangi waktu tunggu dan membagi beban armada.

Keywords— graf berbobot; graf berarah; sirkuit Hamilton; Asymmetric TSP; Nearest Neighbor; Brute Force; rute shuttle

I. PENDAHULUAN

ITB Kampus Jatinangor memiliki populasi mahasiswa yang besar dan tersebar di berbagai kos, apartemen, serta hunian sekitar kampus. Berdasarkan data Direktorat ITB Kampus Jatinangor, terdapat lebih dari 4.481 kamar kos dan apartemen [7] yang tersebar di beberapa kawasan seperti Jl. Caringin, Jl. Kolonel Ahmad Syam, Kampung Geulis, Jatinangor Townsquare, Jl. Cikuda, Jl. Sukawening, Skyland, dan Jl. GKPN. Walaupun jarak tiap kawasan ke kampus tampak relatif

dekat, pola jalan di sekitar Jatinangor membuat jarak tempuh kendaraan tidak selalu sama untuk dua arah perjalanan.

Belum tersedianya layanan shuttle khusus mahasiswa menyebabkan mobilitas harian masih bergantung pada kendaraan pribadi, berjalan kaki, atau ojek daring. Jika setiap mahasiswa melakukan perjalanan secara individual, total biaya dan konsumsi perjalanan menjadi kurang efisien. Permasalahan ini dapat dipandang sebagai persoalan perancangan rute, yaitu kendaraan harus berangkat dari kampus, mengunjungi sejumlah titik kos, lalu kembali ke kampus dengan total jarak sekecil mungkin.

Makalah ini membahas bagaimana kawasan kos mahasiswa ITB Jatinangor dapat dimodelkan sebagai graf berbobot berarah, bagaimana menentukan sirkuit Hamilton minimum untuk rute shuttle, bagaimana perbandingan solusi Nearest Neighbor terhadap solusi optimal Brute Force, serta apakah satu rute tunggal selalu lebih baik daripada dua rute terbagi. Kontribusi makalah ini adalah membangun model ATSP berbasis data jarak berkendara antar kawasan kos, membandingkan rute cepat yang diperoleh dengan Nearest Neighbor dan rute optimal yang diperoleh dengan Brute Force, menganalisis trade-off antara total jarak operasional dan panjang rute per armada, serta menunjukkan pengaruh asimetri jalan terhadap pemilihan model graf.

II. DASAR TEORI

A. Graf dan Graf Berbobot

Graf adalah pasangan himpunan $G = (V, E)$, dengan V sebagai himpunan vertex dan E sebagai himpunan edge [1]. Vertex merepresentasikan objek atau lokasi, sedangkan edge merepresentasikan hubungan antar objek tersebut. Pada graf berbobot, setiap edge diberi nilai tertentu yang disebut bobot. Dalam makalah ini, bobot menyatakan jarak berkendara dalam kilometer.

B. Graf Berarah dan Asimetri Bobot

Graf berarah digunakan ketika hubungan antar vertex memiliki arah [1]. Pada graf berarah, edge dari vertex u ke vertex v tidak selalu sama dengan edge dari vertex v ke vertex u . Dalam konteks jalan raya, jarak dari lokasi A ke lokasi B dapat berbeda dari jarak B ke A karena jalan satu arah, median jalan, atau jalur putar balik. Oleh karena itu, digunakan fungsi bobot $w(u, v)$, yaitu jarak dari vertex u ke vertex v . Jika $w(u, v)$ tidak sama dengan $w(v, u)$, maka graf bersifat asimetris.

C. Lintasan dan Sirkuit Hamilton

Lintasan Hamilton adalah lintasan yang melewati setiap vertex tepat satu kali. Jika lintasan tersebut kembali ke vertex awal, lintasan tersebut disebut sirkuit Hamilton [3]. Dalam konteks shuttle, kampus diperlakukan sebagai titik awal dan akhir rute, sehingga rute yang diinginkan berbentuk ITB $\rightarrow$ titik kos 1 $\rightarrow$ titik kos 2 $\rightarrow \dots \rightarrow$ titik kos $n \rightarrow$ ITB.

D. Travelling Salesman Problem dan ATSP

Travelling Salesman Problem (TSP) adalah persoalan mencari sirkuit Hamilton dengan total bobot minimum [3], [4]. Jika bobot antar dua vertex bergantung pada arah, persoalan ini disebut Asymmetric Travelling Salesman Problem (ATSP). Untuk n vertex dengan depot tetap, jumlah kemungkinan sirkuit berarah adalah $(n - 1)!$. Pada model ini terdapat 9 vertex, sehingga Brute Force mengevaluasi $8! = 40.320$ rute.

E. Algoritma Nearest Neighbor

Nearest Neighbor adalah algoritma greedy yang selalu memilih vertex terdekat yang belum dikunjungi dari posisi saat ini [5]. Algoritma ini memiliki kompleksitas $O(n^2)$, sehingga cepat dan sederhana. Namun, karena hanya mempertimbangkan pilihan lokal, hasilnya tidak dijamin optimal.

F. Algoritma Brute Force

Brute Force mengevaluasi seluruh permutasi vertex selain depot, menghitung total bobot setiap sirkuit, lalu memilih rute dengan bobot terkecil. Kompleksitasnya $O(n!)$, sehingga tidak efisien untuk jumlah vertex besar, tetapi masih layak untuk kasus kecil seperti 9 vertex [4]. Dalam makalah ini, Brute Force digunakan sebagai ground truth untuk menilai kualitas solusi Nearest Neighbor. Ringkasan prinsip dan kompleksitas kedua algoritma yang digunakan ditunjukkan pada Tabel I.

TABEL I. RINGKASAN ALGORITMA YANG DIGUNAKAN

Algoritma	Prinsip	Kompleksitas	Keterangan
Nearest Neighbor	Pilih titik terdekat.	$O(n^2)$	Cepat, tidak selalu optimal.
Brute Force	Coba semua sirkuit.	$O(n!)$	Optimal, mahal.

G. Pengukuran Kualitas Solusi

Misalkan sebuah rute dinyatakan sebagai sirkuit $C = (v_0, v_1, \dots, v_k, v_0)$ yang berawal dan berakhir di depot v_0 . Total bobot rute tersebut diperoleh dengan menjumlahkan bobot seluruh edge berarah yang dilalui [4]:

$$W(C) = \sum_{i=0}^k w(v_i, v_{i+1}), v_{k+1} = v_0 \quad (1)$$

dengan $w(v_i, v_{i+1})$ adalah jarak berkendara dari v_i ke v_{i+1} . Karena graf bersifat berarah, urutan simpul memengaruhi nilai $W(C)$.

Persoalan ATSP bertujuan mencari sirkuit Hamilton C^* dengan total bobot minimum di antara seluruh kemungkinan rute:

$$C^* = \arg \min_{C \in H} W(C) \quad (2)$$

dengan H adalah himpunan semua sirkuit Hamilton berarah yang berawal dan berakhir di depot. Brute Force menyelesaikan persamaan (2) secara eksak dengan mengevaluasi seluruh anggota H , sehingga $W(C^*)$ dipakai sebagai solusi optimal acuan.

Kualitas solusi heuristik diukur melalui gap, yaitu selisih relatif total jarak rute Nearest Neighbor terhadap rute optimal [5]:

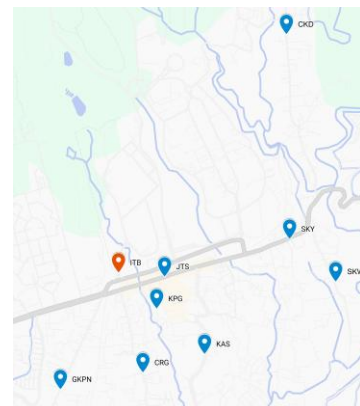
$$\text{Gap} = \frac{W_{\text{heuristik}} - W_{\text{optimal}}}{W_{\text{optimal}}} \times 100\% \quad (3)$$

dengan $W_{\text{heuristik}}$ = total bobot rute Nearest Neighbor dan $W_{\text{optimal}} = W(C^*)$. Gap 0% berarti heuristik mencapai solusi optimal, sedangkan gap yang lebih besar menunjukkan rute heuristik semakin jauh dari optimal.

III. PEMODELAN GRAF DAN METODE

A. Penentuan Vertex

Model graf dibangun dengan menetapkan ITB Jatinangor sebagai depot dan delapan kawasan kos sebagai vertex tujuan. Delapan kawasan dipilih karena mewakili konsentrasi hunian mahasiswa dan tersebar pada sisi barat, selatan, timur, dan utara kampus. Setiap kawasan direpresentasikan oleh satu titik acuan koordinat agar pengukuran jarak konsisten. Persebaran lokasi seluruh vertex ditunjukkan pada Gambar 1 dan daftar lengkap vertex beserta koordinatnya disajikan pada Tabel II.



Gambar 1. Persebaran lokasi ITB Jatinangor dan delapan kawasan kos.

Sumber: Google My Maps, diolah penulis, diakses 19 Juni 2026.

TABEL II. DAFTAR VERTEX GRAF

Kode	Kawasan	Titik koordinat	Dasar pemilihan (kos representatif)
ITB	ITB Jatinangor (Depot)	-6.933749, 107.768362	Titik awal & akhir (kampus)
CRG	Jl. Caringin	-6.940018, 107.769903	Bumi Kartika Asri, Residen 77, Wisma Dara
KAS	Jl. Kolonel Ahmad Syam	-6.938853, 107.773804	Pondok Mutiara, Fadilah
KPG	Kampung Geulis	-6.936009, 107.770775	Wisma Beriman, Burapha Lodge, Askalena
JTS	Jatinangor Townsquare	-6.934010, 107.771272	Apt. Pinewood, Kost Putri Segar, Handayani
CKD	Jl. Cikuda	-6.918809, 107.778995	De Java, Apt. Taman Melati, Casa de Aminda
SKW	Jl. Sukawening	-6.934323, 107.782108	Pondok Kembar, Primadina, Rumah Siswa
SKY	Skyland	-6.931643, 107.779190	Apt. Skyland City
GKPN	Jl. GKPN	-6.941066, 107.764673	Pondok Rumah Bata, Apt. Easton Park

B. Penentuan Edge dan Bobot

Setiap pasangan lokasi dihubungkan oleh edge berarah. Bobot edge menyatakan jarak berkendara dari satu lokasi ke lokasi lain dalam satuan kilometer [6]. Karena jalan di sekitar Jatinangor memiliki beberapa ruas satu arah dan jalur putar balik, jarak dari satu lokasi ke lokasi lain tidak selalu sama dengan jarak sebaliknya.

Graf diasumsikan lengkap karena setiap lokasi dianggap dapat dicapai dari lokasi lain melalui rute jalan yang tersedia. Dengan 9 vertex, graf memiliki $9 \times 8 = 72$ edge berarah non-diagonal. Artinya, setiap vertex memiliki edge menuju delapan vertex lainnya, tetapi bobot edge dari u ke v dapat berbeda dari bobot edge dari v ke u .

Graf lengkap tidak divisualisasikan secara langsung karena jumlah edge berarah mencapai 72, sehingga gambar menjadi terlalu padat. Sebagai gantinya, seluruh edge dan bobotnya direpresentasikan dalam matriks jarak berarah pada Tabel III.

C. Matriks Jarak Berarah

Matriks jarak disusun dengan baris sebagai titik asal dan kolom sebagai titik tujuan [2]. Elemen diagonal bernilai 0 karena menunjukkan jarak dari suatu titik ke dirinya sendiri. Matriks ini menjadi input utama bagi algoritma Nearest Neighbor dan Brute Force.

TABEL III. MATRIKS JARAK ASIMETRIS ANTAR TITIK (KM)

	ITB	CRG	KAS	KPG	JTS	CKD	SKW	SKY	GKPN
ITB	0	3.6	2.1	1	0.6	2.9	2	1.4	1.8
CRG	0.9	0	3	1.8	1.5	3.8	2.8	2.2	1.5
KAS	1.6	1.5	0	1.3	1	3.4	2.5	1.8	2.1
KPG	0.65	4.2	2.7	0	1.2	3.5	2.6	2	1.2
JTS	0.6	4.1	2.7	0.35	0	3.5	2.5	1.9	1.2
CKD	3.1	4.4	3	2.9	2.5	0	1.9	1.6	3.7
SKW	2.1	3.5	2	1.9	1.5	1.9	0	0.6	2.7
SKY	1.5	2.9	1.4	1.3	0.9	1.6	0.6	0	2.1
GKPN	1.9	3.4	4	2.8	2.5	4.8	3.8	3.2	0

Dari matriks tersebut terlihat bahwa sebagian besar pasangan lokasi memiliki jarak yang berbeda untuk dua arah perjalanan. Contohnya, jarak ITB menuju CRG adalah 3,6 km, sedangkan jarak CRG menuju ITB hanya 0,9 km. Hal ini menunjukkan bahwa pemodelan graf berarah diperlukan agar model rute sesuai dengan kondisi jalan sebenarnya.

D. Perancangan Skenario Layanan

Dua skenario layanan diuji dalam makalah ini. Skenario A menggunakan satu rute tunggal yang mengunjungi seluruh kawasan kos dalam satu sirkuit. Skenario ini bertujuan mencari rute dengan total jarak operasional paling kecil.

Skenario B membagi titik tujuan menjadi dua kluster geografis. Rute Selatan-Barat berisi GKPN, CRG, KAS, dan KPG, sedangkan Rute Timur-Utara berisi JTS, SKY, SKW, dan CKD. Kedua rute tetap berangkat dari ITB dan kembali ke ITB. Skenario ini digunakan untuk melihat apakah pembagian rute dapat mengurangi panjang perjalanan per armada dan potensi waktu tunggu mahasiswa.

E. Tahapan Eksperimen

Tahapan eksperimen terdiri atas lima langkah. Pertama, menentukan vertex dan koordinat lokasi. Kedua, menyusun matriks jarak berarah berdasarkan jarak berkendara. Ketiga, menjalankan algoritma Nearest Neighbor dan Brute Force

pada Skenario A. Keempat, menjalankan kedua algoritma pada masing-masing sub-rute Skenario B. Kelima, membandingkan hasil berdasarkan total jarak, gap terhadap solusi optimal, jumlah rute optimal, panjang rute terpanjang per armada, dan interpretasi operasional.

IV. IMPLEMENTASI

A. Struktur Program

Implementasi dilakukan menggunakan program Python. Program menerima matriks jarak sebagai input, lalu menghitung rute shuttle menggunakan dua pendekatan, yaitu Nearest Neighbor dan Brute Force. Matriks jarak yang digunakan merepresentasikan bobot graf berarah, sehingga nilai jarak dari satu titik ke titik lain dapat berbeda dari arah sebaliknya.

Secara umum, program terdiri atas beberapa bagian utama, yaitu pendefinisian vertex dan koordinat, pembacaan matriks jarak, fungsi perhitungan total jarak rute, implementasi Nearest Neighbor, implementasi Brute Force, serta visualisasi rute hasil perhitungan. Output program berupa urutan rute, total jarak, dan gap terhadap solusi optimal.

B. Implementasi Nearest Neighbor

Nearest Neighbor digunakan untuk memperoleh rute secara cepat. Algoritma dimulai dari ITB sebagai titik awal, lalu pada setiap langkah memilih titik terdekat yang belum dikunjungi. Setelah seluruh titik dikunjungi, rute kembali ke ITB. Pseudocode algoritma Nearest Neighbor ditunjukkan pada Gambar 2.

```

1 Algoritma NearestNeighbor
2 { Menghasilkan rute dari titik awal ITB dengan selalu memilih titik
3   terdekat yang belum dikunjungi. }
4
5 KAMUS
6 V      : himpunan vertex
7 D      : matriks jarak
8 R      : list of vertex
9 belum  : himpunan vertex
10 kini   : vertex
11 berikutnya : vertex
12 total  : real
13
14 ALGORITMA
15 R ← [ITB]
16 kini ← ITB
17 belum ← semua vertex selain ITB
18 total ← 0
19
20 while belum ≠ {} do
21   berikutnya ← vertex j dalam belum dengan D[kini, j] minimum
22   total ← total + D[kini, berikutnya]
23   tambahkan berikutnya ke akhir R
24   kini ← berikutnya
25   hapus berikutnya dari belum
26 end while
27
28 total ← total + D[kini, ITB]
29 tambahkan ITB ke akhir R
30
31 return R, total

```

Gambar 2. Pseudocode algoritma Nearest Neighbor.

C. Implementasi Brute Force

Brute Force digunakan untuk memperoleh solusi optimal pada jumlah titik yang masih kecil. Algoritma ini membentuk seluruh permutasi titik selain ITB, lalu menghitung total jarak setiap sirkuit. Rute dengan total jarak terkecil dipilih sebagai

rute optimal. Pseudocode algoritma Brute Force ditunjukkan pada Gambar 3.

```

1 Algoritma BruteForce
2 { Mengevaluasi seluruh permutasi vertex selain ITB untuk memperoleh
3   sirkuit dengan total jarak minimum. Jika terdapat beberapa rute
4   optimal, dipilih rute dengan ruas terpanjang terkecil. }
5
6 KAMUS
7 V      : himpunan vertex
8 D      : matriks jarak
9 titikLain : list of vertex
10 perm    : list of vertex
11 R       : list of vertex
12 r       : list of list of vertex
13 optimal  : list of list of vertex
14 ruteTerpilih : list of vertex
15 total    : real
16 bestTotal : real
17 eps      : real
18
19 ALGORITMA
20 titikLain ← semua vertex selain ITB
21 bestTotal ← tak hingga
22 optimal ← []
23 eps ← 10-9
24
25 for setiap permutasi perm dari titikLain do
26   R ← [ITB] + perm + [ITB]
27   total ← panjangRute(R, D)
28
29   if total < bestTotal - eps then
30     bestTotal ← total
31     optimal ← [R]
32   else if |total - bestTotal| ≤ eps then
33     tambahkan R ke optimal
34   end if
35 end for
36
37 ruteTerpilih ← rute r dalam optimal dengan ruasTerpanjang(r, D) minimum
38
39 return ruteTerpilih, bestTotal, optimal

```

Gambar 3. Pseudocode algoritma Brute Force

D. Tie-Break Rule Optimal

Pada beberapa kasus, terdapat lebih dari satu sirkuit dengan total jarak minimum yang sama. Untuk memilih satu rute representatif, digunakan kriteria tambahan berupa ruas terpanjang terkecil. Artinya, jika terdapat beberapa rute dengan total jarak sama, program memilih rute yang tidak memiliki satu segmen perjalanan yang terlalu panjang.

Kriteria ini relevan untuk konteks shuttle karena rute dengan satu lompatan terlalu jauh dapat membuat distribusi perjalanan kurang seimbang. Dengan tie-break ini, rute yang dipilih tidak hanya minimum secara total, tetapi juga lebih merata antar segmen.

E. Kompleksitas Implementasi

Nearest Neighbor memiliki kompleksitas $O(n^2)$ karena pada setiap langkah algoritma mencari titik terdekat dari himpunan titik yang belum dikunjungi. Brute Force memiliki kompleksitas $O(n!)$ karena harus mengevaluasi seluruh permutasi titik. Pada model ini terdapat 9 vertex, sehingga Brute Force mengevaluasi $8! = 40.320$ sirkuit untuk Skenario A.

Perbandingan kompleksitas ini menunjukkan bahwa Brute Force masih layak digunakan untuk kasus kecil, tetapi tidak cocok untuk jumlah titik yang besar. Sebaliknya, Nearest Neighbor lebih cepat dan mudah diperluas, tetapi hasilnya perlu dibandingkan dengan solusi optimal untuk mengetahui kualitasnya.

V. HASIL DAN ANALISIS

A. Hasil Skenario A: Rute Tunggal

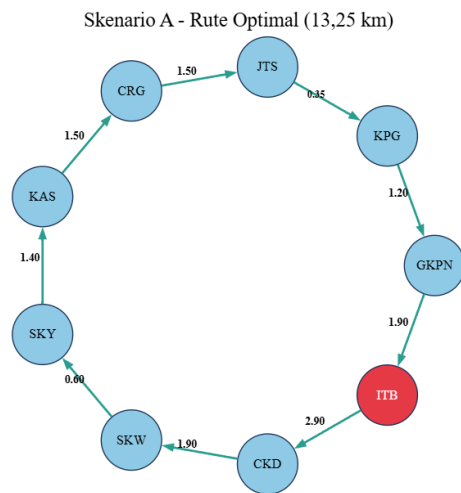
Pada Skenario A, seluruh vertex dilayani dalam satu sirkuit. Nearest Neighbor menghasilkan rute:

ITB → JTS → KPG → GKPN → SKY → SKW → CKD → KAS → CRG → ITB

Total jaraknya adalah 13,25 km. Brute Force menghasilkan salah satu rute optimal:

ITB → CKD → SKW → SKY → KAS → CRG → JTS → KPG → GKPN → ITB

Total jaraknya juga 13,25 km. Dengan demikian, pada Skenario A, Nearest Neighbor mencapai solusi optimal dengan gap 0%. Namun, hasil ini tidak berarti Nearest Neighbor selalu optimal. Eksperimen juga menemukan enam sirkuit berbeda dengan total minimum 13,25 km, sehingga solusi optimal pada kasus ini tidak tunggal. Sirkuit Hamilton optimal Skenario A ditunjukkan pada Gambar 4.



Gambar 4. Sirkuit Hamilton optimal Skenario A dengan total jarak 13,25 km

B. Hasil Skenario B: Rute Terbagi

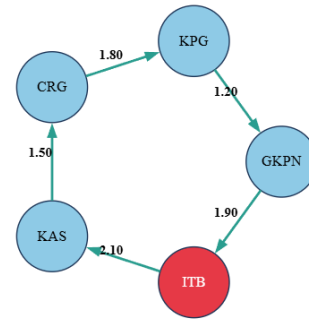
Pada Skenario B, titik dibagi menjadi dua rute berdasarkan kedekatan geografis. Rute pertama adalah Rute Selatan-Barat yang meliputi GKPN, CRG, KAS, dan KPG. Rute kedua adalah Rute Timur-Utara yang meliputi JTS, SKY, SKW, dan CKD.

Untuk Rute Selatan-Barat, Brute Force menghasilkan rute optimal:

ITB → KAS → CRG → KPG → GKPN → ITB

Total jaraknya adalah 8,50 km. Sementara itu, Nearest Neighbor menghasilkan rute dengan total 10,20 km, sehingga gap terhadap optimal adalah 20,00%. Rute optimal Selatan-Barat ditunjukkan pada Gambar 5.

Skenario B1 - Rute Selatan-Barat (8,50 km)



Gambar 5. Sirkuit Hamilton optimal Skenario B pada Rute Selatan-Barat

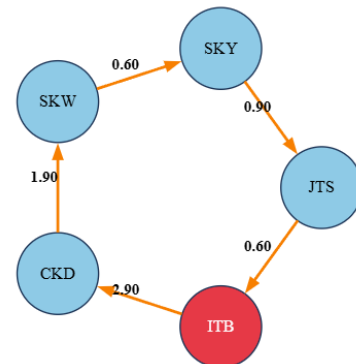
Total jaraknya adalah 8,50 km. Sementara itu, Nearest Neighbor menghasilkan rute dengan total 10,20 km, sehingga gap terhadap optimal adalah 20,00%. Rute optimal Selatan-Barat ditunjukkan pada Gambar 5.

Untuk Rute Timur-Utara, Brute Force menghasilkan rute optimal:

ITB → CKD → SKW → SKY → JTS → ITB

Total jaraknya adalah 6,90 km. Sementara itu, Nearest Neighbor menghasilkan rute dengan total 8,10 km, sehingga gap terhadap optimal adalah 17,39%. Rute optimal Timur-Utara ditunjukkan pada Gambar 6.

Skenario B2 - Rute Timur-Utara (6,90 km)



Gambar 6. Sirkuit Hamilton optimal Skenario B pada Rute Timur-Utara

Total jarak gabungan Skenario B adalah 15,40 km. Nilai ini lebih besar daripada Skenario A, tetapi masing-masing rute pada Skenario B menjadi lebih pendek jika dijalankan oleh armada yang berbeda.

C. Perbandingan Nearest Neighbor dan Brute Force

Hasil eksperimen menunjukkan bahwa Nearest Neighbor dapat menghasilkan solusi optimal pada kasus tertentu, tetapi performanya sangat bergantung pada struktur bobot graf. Pada Skenario A, pilihan lokal yang diambil Nearest Neighbor kebetulan membentuk rute dengan total minimum. Sebaliknya, pada dua sub-rute Skenario B, pilihan lokal awal menyebabkan rute menjadi lebih panjang daripada optimal. Perbandingan hasil kedua algoritma pada tiap skenario dirangkum pada Tabel IV.

TABEL IV. PERBANDINGAN HASIL DAN ALGORITMA

Skenario	Metode	Rute	Total Jarak	Gap
A	Nearest Neighbor	ITB-JTS-KPG-GKPN-SKY-SKW-CKD-KAS-CRG-ITB	13,25 km	0%
A	Brute Force	ITB-CKD-SKW-SKY-KAS-CRG-JTS-KPG-GKPN-ITB	13,25 km	0%
B Selatan-Barat	Nearest Neighbor	ITB-KPG-GKPN-CRG-KAS-ITB	10,20 km	20,00%
B Selatan-Barat	Brute Force	ITB-KAS-CRG-KPG-GKPN-ITB	8,50 km	0%
B Timur-Utara	Nearest Neighbor	ITB-JTS-SKY-SKW-CKD-ITB	8,10 km	17,39%
B Timur-Utara	Brute Force	ITB-CKD-SKW-SKY-JTS-ITB	6,90 km	0%

D. Pengaruh Asimetri Jarak

Asimetri bobot merupakan alasan utama mengapa model ATSP lebih tepat daripada TSP simetris. Dari 36 pasangan lokasi tidak berarah, 32 pasangan memiliki jarak dua arah yang berbeda, atau 88,9%. Rata-rata selisih absolut antar arah adalah sekitar 0,91 km. Selisih terbesar terjadi pada pasangan ITB dan Caringin: ITB → CRG sejauh 3,6 km, sedangkan CRG → ITB hanya 0,9 km.

Jika arah diabaikan, rute yang dihasilkan dapat terlihat optimal di atas kertas tetapi tidak merepresentasikan perjalanan kendaraan sebenarnya. Oleh sebab itu, penggunaan graf berarah bukan hanya pilihan formal, tetapi kebutuhan agar model sesuai dengan kondisi jalan nyata.

E. Perbandingan Skenario A dan Skenario B

Skenario A memiliki total jarak operasional terkecil, yaitu 13,25 km. Skenario B memiliki total gabungan 15,40 km, atau 16,23% lebih panjang. Jika satu-satunya tujuan adalah minimisasi biaya bahan bakar atau total jarak armada, Skenario A lebih unggul.

Namun, jika layanan dijalankan secara paralel oleh dua armada, Skenario B memiliki keunggulan operasional. Rute terpanjang per armada turun dari 13,25 km menjadi 8,50 km, yaitu penurunan sekitar 35,85%. Dengan asumsi kecepatan rata-rata kendaraan sama, penurunan panjang rute terpanjang dapat mengurangi waktu tunggu mahasiswa yang berada di titik akhir rute. Pada asumsi sederhana kecepatan rata-rata 20 km/jam, durasi rute tunggal sekitar 39,75 menit, sedangkan rute terpanjang pada Skenario B sekitar 25,50 menit. Ringkasan perbandingan kedua skenario disajikan pada Tabel V.

TABEL V. PERBANDINGAN SKENARIO LAYANAN

Aspek	Skenario A	Skenario B
Jumlah rute	1 rute tunggal	2 rute terbagi
Total jarak	13,25 km	15,40 km
Rute terpanjang per armada	13,25 km	8,50 km
Kelebihan	Total jarak paling kecil	Waktu tunggu lebih rendah, kapasitas lebih fleksibel
Kekurangan	Satu armada menempuh rute panjang	Total jarak gabungan lebih besar
Cocok jika prioritas	Efisiensi biaya/jarak	Kapasitas dan waktu tunggu

F. Keterbatasan Model dan Rekomendasi

Model ini masih memiliki beberapa keterbatasan. Pertama, bobot hanya menggunakan jarak, belum waktu tempuh aktual. Kedua, setiap kawasan kos direpresentasikan oleh satu titik, padahal mahasiswa tersebar di banyak bangunan. Ketiga, permintaan penumpang dan kapasitas kendaraan belum dimodelkan secara kuantitatif. Keempat, pembagian klaster pada Skenario B masih berbasis kedekatan geografis, belum dioptimalkan dengan algoritma clustering.

Pengembangan selanjutnya dapat menggunakan bobot waktu tempuh, data permintaan mahasiswa, kapasitas kendaraan, serta algoritma tambahan seperti Cheapest Link, 2-opt, atau metaheuristik untuk kasus dengan vertex lebih banyak. Model juga dapat diperluas menjadi Vehicle Routing Problem (VRP) agar lebih mendekati masalah operasional shuttle sebenarnya.

VI. KESIMPULAN

Makalah ini berhasil memodelkan rute shuttle mahasiswa ITB Jatnangor sebagai graf berbobot berarah dengan sembilan vertex. Karena 88,9% pasangan lokasi memiliki jarak dua arah yang berbeda, penggunaan graf berarah dan pendekatan Asymmetric Travelling Salesman Problem lebih tepat dibandingkan graf tak berarah.

Pada Skenario A, rute optimal memiliki total jarak 13,25 km. Nearest Neighbor menghasilkan total yang sama dengan Brute Force sehingga gap-nya 0%, tetapi hasil pada Skenario B menunjukkan bahwa metode greedy ini tidak selalu optimal. Pada Skenario B, total jarak gabungan optimal adalah 15,40 km, dengan gap Nearest Neighbor sebesar 20,00% pada Rute Selatan-Barat dan 17,39% pada Rute Timur-Utara.

Skenario A unggul dalam minimisasi total jarak, sedangkan Skenario B unggul dalam pembagian beban armada dan potensi pengurangan waktu tunggu. Dengan demikian, keputusan rute shuttle tidak cukup hanya berdasarkan jarak total, tetapi juga perlu mempertimbangkan kapasitas, waktu tunggu, dan tujuan layanan. Penelitian lanjutan dapat memasukkan data waktu tempuh aktual, jam

sibuk, permintaan mahasiswa, dan kapasitas kendaraan agar model semakin mendekati kebutuhan operasional shuttle nyata.

LAMPIRAN

Implementasi program (program_sirkuit.py), berkas matriks jarak (jarak_antar_titik.csv), peta lokasi seluruh vertex (peta_lokasi_vertex.pdf), data 72 edge berarah beserta tautan Google Maps dan tanggal akses (data_jarak_dan_referensi.xlsx), gambar visualisasi rute tiap skenario, serta tangkapan keluaran program tersedia pada folder berikut: https://drive.google.com/drive/folders/18nRkUaC-V_T-9MrK5FyrEY1rXSge3SME?usp=sharing

LINK VIDEO YOUTUBE

<https://youtu.be/44kl6kNOz1o?si=Vp3seDTtJeWnd6Gg>

UCAPAN TERIMA KASIH

Pertama-tama, penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga makalah ini dapat diselesaikan. Penulis juga mengucapkan terima kasih kepada Prof. Dr. Ir. Rinaldi, M.T. selaku dosen pengampu IF1220 Matematika Diskrit atas materi, arahan, dan referensi perkuliahan yang menjadi dasar penyusunan makalah ini. Tidak lupa, penulis berterima kasih kepada keluarga dan rekan-rekan yang telah memberikan dukungan selama proses penulisan makalah.

REFERENSI

- [1] R. Munir, "Graf (Bagian 1)," Bahan Kuliah IF1220 Matematika Diskrit, Institut Teknologi Bandung, 2026. [Daring]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf> [Diakses: 14 Juni 2026, 16.00 WIB].
- [2] R. Munir, "Graf (Bagian 2)," Bahan Kuliah IF1220 Matematika Diskrit, Institut Teknologi Bandung, 2026. [Daring]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/21-Graf-Bagian2-2026.pdf> [Diakses: 14 Juni 2026, 16.00 WIB].
- [3] R. Munir, "Graf (Bagian 3)," Bahan Kuliah IF1220 Matematika Diskrit, Institut Teknologi Bandung, 2026. [Daring]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/22-Graf-Bagian3-2026.pdf> [Diakses: 14 Juni 2026, 16.00 WIB].
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, dan C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA, USA: The MIT Press, 2022.
- [5] D. J. Rosenkrantz, R. E. Stearns, dan P. M. Lewis II, "An Analysis of Several Heuristics for the Traveling Salesman Problem," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 563–581, 1977, doi: 10.1137/0206041.

[6] Google, "Google Maps," [Daring]. Tersedia: <https://www.google.com/maps> [Diakses: 15 Juni 2026, 20.00 WIB].

[7] Direktorat ITB Kampus Jatinangor, Seksi Humas, "Ketersediaan Kost di Jatinangor guna Memenuhi Kebutuhan Tempat Tinggal Mahasiswa," Institut Teknologi Bandung, Jatinangor, Januari 2023.

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.

Jakarta, 19 Juni 2026



Avicenna Ananda Musthafa
13525018